Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

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 №3347

очного отделения:

Якимов Ярослав

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Федоров И.Р.



Санкт-Петербург, 2020

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

**Задачи:**

* Подсчет стоимости всех услуг клиента
* Формирование счета

**Средства реализации:** для реализации был выбран язык программирования Python, как отвечающий всем современным требованиям в разработке. Кроме того, он предоставляет возможность написать быстрый код, как по времени разработки, так и по времени исполнения. Тем более он имеет встроенные модули для работы с файлами json и библиотеки для формирования удобных и читабельных графиков – docxtpl и docx2pdf.

Исходный код (так же по [ссылке](https://github.com/coltmister/MobileDevYakimov)):

import csv

import json

import time

from docxtpl import DocxTemplate

from docx2pdf import convert

def celluar(data, msisdn\_origin, in\_cost, out\_cost, sms\_cost, first\_sms=0, first\_minutes\_out=0, first\_minutes\_in=0, first\_sms\_cost=0, first\_minutes\_out\_cost=0,

first\_minutes\_in\_cost=0):

minutes\_result = 0

sms\_result = 0

for row in data:

# Если это исходящие звонки, то тарификация по исходящим

if msisdn\_origin == row['msisdn\_origin']:

minutes\_result += (float(

row['call\_duration']) - first\_minutes\_out) \* out\_cost + first\_minutes\_out \* first\_minutes\_out\_cost

# Считаем сколько стоят исходящие минуты, и т.к. "первые" минуты считаются по звонку, то мы их не сбрасываем

if first\_sms != 0: # смс сбрасываются, так что

if first\_sms < int(row['sms\_number']):

sms\_result += (int(row['sms\_number']) - first\_sms) \* \

sms\_cost + first\_sms \* first\_sms\_cost

first\_sms = 0

else:

first\_sms -= int(row['sms\_number'])

else:

sms\_result += (int(row['sms\_number'])) \* sms\_cost

if msisdn\_origin == row['msisdn\_dest']: # Входящие звонки

minutes\_result += (float(row['call\_duration']) - first\_minutes\_in) \* \

in\_cost + first\_minutes\_in \* first\_minutes\_in\_cost

return minutes\_result, sms\_result

def internet(person\_ip, data, first\_price, second\_price):

volume = 0

for item in data:

if 'src4\_addr' in item: # Так как нам тут не надо знать значения дат и трафик считается по одинаковым ценам

# То мы просто суммируем в одну переменную

if item['src4\_addr'] == person\_ip or item['dst4\_addr'] == person\_ip:

volume += int(item['in\_bytes'])

sum\_traffic = (volume) / 1024 # Кб

if sum\_traffic > 200:

result = (sum\_traffic - 200) \* first\_price + 200 \* second\_price

else:

result = sum\_traffic\*second\_price

return result

def main():

with open("context.json", 'r', encoding='utf-8') as f:

context = json.load(f)

with open('dump.json', "r") as f:

dump = json.load(f)

data = csv.DictReader(open('data.csv'))

minutes\_total, sms\_total = celluar(

data, '933156729', 0, 2, 1, first\_sms=10)

internet\_total = internet("192.0.73.2", dump, 1, 0.5)

context['bill']['content'].extend([

{

"id": "1",

"description": "Интернет трафик",

"quantity": "",

"unit": "",

"price": f'{internet\_total:.2f}'.replace('.', ', '),

"amount": f'{internet\_total:.2f}'.replace('.', ', ')

},

{

"id": "2",

"description": "Мобильная связь, включая СМС и Звонки",

"quantity": "",

"unit": "",

"price": f'{minutes\_total+sms\_total:.2f}'.replace('.', ', '),

"amount": f'{minutes\_total+sms\_total:.2f}'.replace('.', ', ')

},

{

"id": "2.1",

"description": "Звонки",

"quantity": "",

"unit": "",

"price": f'{minutes\_total:.2f}'.replace('.', ', '),

"amount": f'{minutes\_total:.2f}'.replace('.', ', ')

},

{

"id": "2.2",

"description": "СМС",

"quantity": "",

"unit": "",

"price": f'{sms\_total:.2f}'.replace('.', ', '),

"amount": f'{sms\_total:.2f}'.replace('.', ', ')

}])

context['bill']['total'] = f'{(sms\_total + minutes\_total + internet\_total):.2f}'.replace(

'.', ', ')

context['bill']['taxes'] = f'{(0.2 \* (sms\_total + minutes\_total + internet\_total)):.2f}'.replace('.', ', ')

print(

f"Введите число прописью, я сам не умею пока: {context['bill']['total']}")

context['bill']['total\_in\_words'] = input("Ваш ответ: ")

context['bill']['total'] = f'{(sms\_total + minutes\_total + internet\_total):.2f}'.replace(

'.', ', ')

doc = DocxTemplate("template.docx")

doc.render(context)

name = f"Счет №{context['bill']['id']}.docx"

doc.save(name)

convert(name)

start\_time = time.time()

main()

print(f'Took {time.time() - start\_time} seconds')

**Выводы:** Я научился автоматизировать процесс создания документов для компании.