

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:

03.04.20 Якимов Ярослав



Проверил

Федоров И.Р.



УНИВЕРСИТЕТ ИТМО

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](#)):

```
import csv
import json
import time

from docxtpl import DocxTemplate
from docx2pdf import convert

def cellular(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.