**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Национальный исследовательский университет ИТМО»**

**Факультет безопасности информационных технологий**

**Дисциплина:**

**«Управление мобильными устройствами»**

**Лабораторная работа № 3 на тему**

**«Формирование счета на оплату услуг»**

**Выполнил:**

Студент гр. N3349

Будянский Д.С.

**Проверил:**

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

Федоров И. Р.

Санкт-Петербург

2020г.

**Цель работы:** По полученным результатам тарификации услуг «Телефония» и «Интернет» в лабораторных работах 1, 2 сформировать счет на оплату в формате .pdf

**Исходные файлы**: «data.csv», «datai.csv».

**Ход работы**:

data.csv - исходный файл ЛР 1, datai.csv - результат обработки исходного файла ЛР 2.

Для обработки исходного файла в ЛР 2 использовалась команда   
«nfdump -r nfcapd.202002251200 -o csv> datai.csv», сохраняющая csv вид данного файла в файл «datai.csv».

Для написания ПО для тарификации был выбран язык программирования python (версия 3.8). Используемые библиотеки: csv, fpdf, os.

**Входные данные ПО**: data.csv, datai.csv и папка Font\_Dejavu\_spacebefore со шрифтами для печати символов кириллицы («DejaVuSansCondensed.ttf», «DejaVuSansCondensed-Bold.ttf» внутри папки).

**Выходные данные ПО**: «Bill.pdf» - готовый счет за услуги.

**Исходный код:**

import csv

import fpdf

import os

# 1 Лабораторная

msisdn\_origin = []

msisdn\_dest = []

call\_duration = []

sms\_number = []

number = '968247916'

call\_vx\_cost = 1

call\_isx\_cost = 3

sms\_cost = 1

summa\_vx = 0

summa\_isx = 0

summa\_sms = 0

summa\_all = 0

# сохраняем из csv файла данные нужных столбцов(всех кроме timestamp)

with open('data.csv', "r", newline="") as file:

reader = csv.DictReader(file, delimiter=',')

for row in reader:

cur\_arr = row['msisdn\_origin']

msisdn\_origin.extend([cur\_arr])

cur\_arr = row['msisdn\_dest']

msisdn\_dest.extend([cur\_arr])

cur\_arr = float(row['call\_duration'])

call\_duration.extend([cur\_arr])

cur\_arr = int(row['sms\_number'])

sms\_number.extend([cur\_arr])

# ищем нужный нам по варианту телефонный номер и рассчитываем стоимость различных услуг

i = 0

while i < len(msisdn\_dest):

if msisdn\_dest[i] == number:

summa\_vx = summa\_vx + call\_duration[i]

i += 1

summa\_vx = summa\_vx \* call\_vx\_cost

i = 0

while i < len(msisdn\_origin):

if msisdn\_origin[i] == number:

summa\_isx = summa\_isx + call\_duration[i]

i += 1

summa\_isx = summa\_isx \* call\_isx\_cost

i = 0

while i < len(sms\_number):

if msisdn\_origin[i] == number:

summa\_sms = summa\_sms + sms\_number[i]

i += 1

summa\_sms = summa\_sms \* sms\_cost

# подсчитываем общую сумму

summa\_all = summa\_sms + summa\_vx + summa\_isx

# 2 Лабораторная

ip = '217.15.20.194'

traffic\_price = 1

# 1000 Kb бесплатно т.к. требуется тарификация после 1000Мб, а у абонента меньше 1000

free\_traffic = 1

with open('datai.csv', "r", newline="") as file:

reader = csv.DictReader(file, delimiter=',')

summ\_bits = 0

to\_check = []

# Обрабатываем входные данные

for row in reader:

if row['sa'] == ip or row['da'] == ip:

bits\_length = int(row['ibyt'])

summ\_bits += bits\_length

to\_check.append(('Income:' if row['sa'] == ip else 'Outcome:') + f'{bits\_length} bytes')

# Тарифицируем абонента

summ\_mb = summ\_bits / 2 \*\* 20

price = (summ\_mb - free\_traffic) \* traffic\_price if summ\_mb > free\_traffic else 0

summa\_vsego = price + summa\_all

# создание пдф файла

def to\_pdf(name\_of\_file, name\_of\_bank, inn, kpp, bik, recipient, num1, num2,

bill\_number, bill\_date, provider, customer, cause, summa\_vsego):

# Массив для заполнения таблицы со списком услуг и их цен

data = [['Входящие звонки', str(summa\_vx / call\_vx\_cost), 'Минуты', str(call\_vx\_cost), str(summa\_vx)],

['Исходящие звонки', str(summa\_isx / call\_isx\_cost), 'Минуты', str(call\_isx\_cost), str(summa\_isx)],

['СМС', str(summa\_sms / sms\_cost), 'Штуки', str(sms\_cost), str(summa\_sms)],

['Сетевой трафик', str('%.2f' % summ\_mb), 'Мегабайты', str(traffic\_price), str('%.2f' % price)]]

pdf = fpdf.FPDF()

pdf.set\_right\_margin(15)

pdf.set\_left\_margin(15)

pdf.add\_page()

# Добавляем шрифт

pdf.add\_font('DejaVu', '', os.path.join('.', 'Font\_Dejavu\_spacebefore', 'DejaVuSansCondensed.ttf'), uni=True)

pdf.add\_font('DejaVu', 'B', os.path.join('.', 'Font\_Dejavu\_spacebefore', 'DejaVuSansCondensed-Bold.ttf'), uni=True)

pdf.set\_font('DejaVu', '', 10)

headers = [[f'Банк получателя: {name\_of\_bank}', f'БИК: {bik}'],

[f'ИНН: {inn}', f'КПП: {kpp}', f'Сч. № {num1}'],

[f'Получатель: {recipient}', f'Сч. № {num2}']]

col\_width = (pdf.w - 30) / 2

row\_height = pdf.font\_size \* 2

for rows in headers:

for col in rows:

pdf.cell(col\_width / 2 if 'ИНН' in col or 'КПП' in col else col\_width, row\_height,

txt=col, border=1)

pdf.ln(row\_height)

pdf.set\_font('DejaVu', 'B', 14)

s = f'Счёт №{bill\_number} от {bill\_date} г.'

margins = int((pdf.w - pdf.get\_string\_width(s) - 30) / 2 / pdf.get\_string\_width(' ')) \* ' '

pdf.write(14, margins + s)

pdf.ln(5)

pdf.write(14, '\_' \* int((pdf.w - 30) / pdf.get\_string\_width('\_')))

pdf.set\_font('DejaVu', '', 10)

pdf.ln(10)

pdf.write(10, 'Поставщик')

pdf.ln(5)

pdf.write(10, f'(Исполнитель): {provider}')

pdf.ln(10)

pdf.write(10, 'Покупатель')

pdf.ln(5)

pdf.write(10, f'(Заказчик): {customer}')

pdf.ln(10)

pdf.write(10, f'Основание: {cause}')

pdf.ln(10)

row\_height = pdf.font\_size \* 2

pdf.cell(10, row\_height, txt='№', border=1) # №

pdf.cell(70, row\_height, txt='Наименование работ, услуг', border=1) # Наименование

pdf.cell(15, row\_height, txt='Кол-вo', border=1) # Кол-во

pdf.cell(30, row\_height, txt='Ед.', border=1) # Ед.

pdf.cell(25, row\_height, txt='Цена', border=1) # Ценна

pdf.cell(25, row\_height, txt='Сумма', border=1) # Сумма\*

pdf.ln(row\_height)

for rows in data:

row\_num = 0

pdf.cell(10, row\_height, txt=str(row\_num + 1), border=1) # №

pdf.cell(70, row\_height, txt=rows[0], border=1) # Наименование

pdf.cell(15, row\_height, txt=rows[1], border=1) # Кол-во

pdf.cell(30, row\_height, txt=rows[2], border=1) # Ед.

pdf.cell(25, row\_height, txt=rows[3], border=1) # Ценна

pdf.cell(25, row\_height, txt=rows[4], border=1) # Сумма\*

pdf.ln(row\_height)

summa\_vsego = float("{0:.2f}".format(summa\_vsego))

strings = [f'Итого: {summa\_vsego } руб.', f'В том числе НДС: 0 руб.', f'Всего к оплате: {summa\_vsego} руб.']

pdf.set\_font('DejaVu', 'B', 10)

for s in strings:

margins = int((pdf.w - pdf.get\_string\_width(s) - 36) / pdf.get\_string\_width(' ')) \* ' '

pdf.write(10, margins + s)

pdf.ln(5)

pdf.ln(5)

pdf.set\_font('DejaVu', '', 10)

strings = ['\*Сумма была расчитана с учетом цены и количества первых бесплатных единиц по вашему тарифу', 'Внимание!', 'Оплата данного счета означает согласие с условиями поставки товара.',

'Уведомление об оплате обязательно, в противном случае не гарантируется наличие товара на складе.',

'Товар отпускается по факту прихода денег на р/с Поставщика, самовывозом, при наличии',

'доверенности и паспорта.']

for s in strings:

pdf.write(10, s)

pdf.ln(5)

pdf.set\_font('DejaVu', 'B', 14)

pdf.write(14, '\_' \* int((pdf.w - 30) / pdf.get\_string\_width('\_')))

pdf.ln(20)

pdf.set\_font('DejaVu', '', 10)

margins = int((pdf.w - pdf.get\_string\_width('РуководительБухгалтер') - 30) / 2 / pdf.get\_string\_width('\_')) \* '\_'

pdf.write(10, f'Руководитель{margins}Бухгалтер{margins}')

pdf.output(name=name\_of\_file)

to\_pdf('Bill.pdf', 'MMMS', '316451354', '3451', '3154', 'Пржевальскый', '1254', '2245',

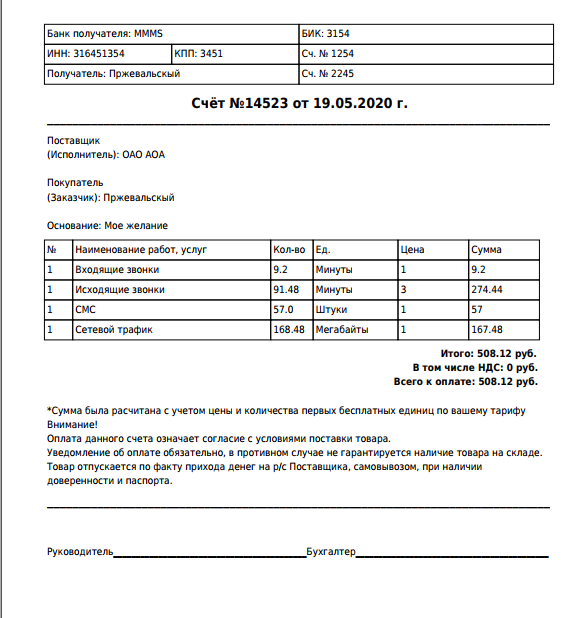
'14523', '19.05.2020', 'ОАО АОА', 'Пржевальскый', 'Мое желание', summa\_vsego)

print('Счет был успешно сформирован.')

**Вывод программы**:



Получившийся счет:



**Вывод**: в результате работы была написана программа, позволяющая по полученным результатам тарификации услуг «Телефония» и «Интернет» в лабораторных работах 1, 2 сформировать счет на оплату в формате .pdf