**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

Шишков В. Е.

**Проверил:**

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

Федоров И. Р.

Санкт-Петербург

2020г.

**Цель работы:** создать программное обеспечение для автоматизации процесса формирования счета за услуги (звонки, смс, интернет), тарификация которых рассчитывалась в лабораторной работе № 1 и № 2.

**Исходный файлы**: «data.csv», «dump.csv».

**Ход работы**:

Исходные файлы были взяты из предыдущих лабораторных работ («data.csv» - исходный файл ЛР 1, «dump.csv» - результат обработки исходного файла ЛР 2).

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

Для написания ПО для тарификации был выбран язык программирования python (версия 3.6.9). Используемые библиотеки: csv, fpdf, os.

Основой выбора данного языка программирования является: простота понимания программного кода и мгновенная коррекция в случае изменения условий тарификации; наличие библиотек для обработки файлов csv формата и создания файлов pdf формата.

**Входные данные ПО**: «data.csv», «dump.csv» и папка «fonts\_for\_fpdf» со шрифтами для печати символов кириллицы (а именно файлы «DejaVuSansCondensed.ttf», «DejaVuSansCondensed-Bold.ttf» внутри папки).

**Выходные данные ПО**: «lab3\_output.pdf» - готовый счет за услуги.

**Исходный код:**

#!/usr/bin/env python3

import csv

import fpdf

import os

def tariff\_calls(data, number, incoming\_calls\_cost, outgoing\_calls\_cost, free\_sms, sms\_cost):

total = 0

stats = []

for row in data:

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

# Обработка исходящих вызовов

duration = float(row['call\_duration'])

cost = duration \* outgoing\_calls\_cost

total += cost

stats.append(('Исходящие звонки', str(round(duration, 1)), 'Минуты', str(outgoing\_calls\_cost), str(round(cost, 1))))

# Обработка смс сообщений

sms\_num = int(row['sms\_number'])

cost = (sms\_num - free\_sms) \* sms\_cost if sms\_num > free\_sms else 0

total += cost

stats.append(('СМС', str(sms\_num), 'Штук', str(sms\_cost), str(round(cost, 1))))

if row['msisdn\_dest'] == number:

# Обработка входящих вызовов

duration = float(row['call\_duration'])

cost = duration \* incoming\_calls\_cost

total += cost

stats.append(('Входящие звонки', str(round(duration, 1)), 'Минуты', str(incoming\_calls\_cost), str(round(cost, 1))))

return stats, total

def tariff\_internet(data, ip\_addr, traffic\_cost, free\_traffic):

total\_bytes = 0

for row in data:

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

# Обработка входящих и исходящих пакетов

bytes\_len = int(row['ibyt'])

total\_bytes += bytes\_len

total\_mb = total\_bytes / 2 \*\* 20

total = round((total\_mb - free\_traffic) \* traffic\_cost, 1) if total\_mb > free\_traffic else 0

return ('Интернет трафик', str(round(total\_mb, 1)), 'Мб', str(traffic\_cost), str(total)), total

def to\_pdf(filename, bank\_name, inn, kpp, bik, recipient, acc\_num1, acc\_num2, \

bill\_num, bill\_date, provider, customer, cause, \

stats, total):

pdf = fpdf.FPDF()

pdf.set\_right\_margin(15)

pdf.set\_left\_margin(15)

pdf.add\_page()

# adding fonts for russian letters

pdf.add\_font('DejaVu', '', os.path.join('.', 'fonts\_for\_fpdf', 'DejaVuSansCondensed.ttf'), uni=True)

pdf.add\_font('DejaVu', 'B', os.path.join('.', 'fonts\_for\_fpdf', 'DejaVuSansCondensed-Bold.ttf'), uni=True)

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

headers = [[f'Банк получателя: {bank\_name}', f'БИК: {bik}'],

[f'ИНН: {inn}', f'КПП: {kpp}', f'Сч. № {acc\_num1}'],

[f'Получатель: {recipient}', f'Сч. № {acc\_num2}']]

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

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

for row in headers:

for col in row:

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\_num} 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 row\_num, row in enumerate(stats):

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

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

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

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

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

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

pdf.ln(row\_height)

strings = [f'Итого: {total} руб.', f'В том числе НДС: 0 руб.', f'Всего к оплате: {total} 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 = ['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=filename)

if \_\_name\_\_ == '\_\_main\_\_':

stats = []

# Data from first lab

with open('data.csv', 'r') as csvfile:

file = list(csv.DictReader(csvfile))

stats\_temp, calls\_total = tariff\_calls(file, '915642913', 1, 1, 5, 1)

stats.extend(stats\_temp)

# Data from second lab

with open('dump.csv', 'r') as csvfile:

file = list(csv.DictReader(csvfile))

stats\_temp, internet\_total = tariff\_internet(file, '192.168.250.59', 1, 1)

stats.append(stats\_temp)

to\_pdf('lab3\_output.pdf', 'Имя банка', 'ИНН', 'КПП', 'БИК', 'Иванов', '1', '2',\

'1', '9.04.2020', 'ОАО Провайдер', 'Иванов', 'Стандартная подписка',\

stats, calls\_total + internet\_total)

**Вывод программы**:

**Вывод**: в результате работы была написана программа, позволяющая сформировать счет на основе записей о звонках, смс сообщениях и интернет трафике абонента.