	<p>Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)</p>
---	--

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКЕ

Студент **Леонов Владислав Вячеславович**

Группа **ИУ7-16Б**

Тип практики **Распределенная**

Название предприятия **МГТУ им. Н.Э. Баумана**

Студент	_____	Леонов В.В.
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Руководитель практики	_____	Кузнецова О. В.
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Оценка _____

2019 г.

Оглавление

Индивидуальное задание.....	3
Описание. Телефонная неразбериха	3
Входные данные	3
Выходные данные	3
Введение.....	4
Основная часть	5
Блок схема	5
Описание программы	6
Листинг	7
Тесты	9
Тест 1.	9
Тест 2.	9
Тест 3.	10
Тест 4.	10
Заключение	11

Индивидуальное задание

Описание. Телефонная неразбериха

Крупной компании необходимо отобразить список счетов за звонки, сделанные ее сотрудниками. У компании имеется информация о номерах телефонов (строка размером до 15 символов), по которым сделаны звонки и о продолжительности разговоров. Написать программу для обработки этих данных и вывода информации о каждом звонке и его цене в формате стандартных телефонных счетов.

Международные (МНК) номера начинаются с двух нулей (00), за которыми следует код страны (1-3 позиции) и номер абонента (4-10 позиций). Международные (МГК) номера начинаются с нуля (0), за которым следует код региона (1-5 позиций) и номер абонента (4-7 позиций). Цена звонка определяется продолжительностью разговора и местом назначения. Звонки в пределах города начинаются с любой отличной от нуля цифры. Эти звонки бесплатны.

Входные данные

Входные данные состоят из двух частей. Первая часть содержит таблицу МНК и МГК кодов, пунктов назначения и цен как показано ниже:

Код ΔПункт назначения\$Цена в центах за минуту,

где Δ – пробел. Пункт назначения – строка длиной до 25 символов. Эта часть входных данных заканчивается строкой из 6 нулей (000000).

Вторая часть содержит серию строк (по одной для каждого звонка), в каждой из которых находятся номер телефона и продолжительность разговора. Файл заканчивается строкой из одного символа #. Данные в этой части не обязательно должны быть табулированы, хотя между ними должен быть по меньшей мере один пробел. Телефонные номера не могут повторяться.

Выходные данные

Выходные данные состоят из номера телефона, пункта назначения, номера абонента, продолжительности разговора, цены за минуту разговора и стоимости разговора. Если номер телефона содержит неопознанный код, в пункте назначения записать „Неизвестно“, а стоимость такого звонка -1.00.

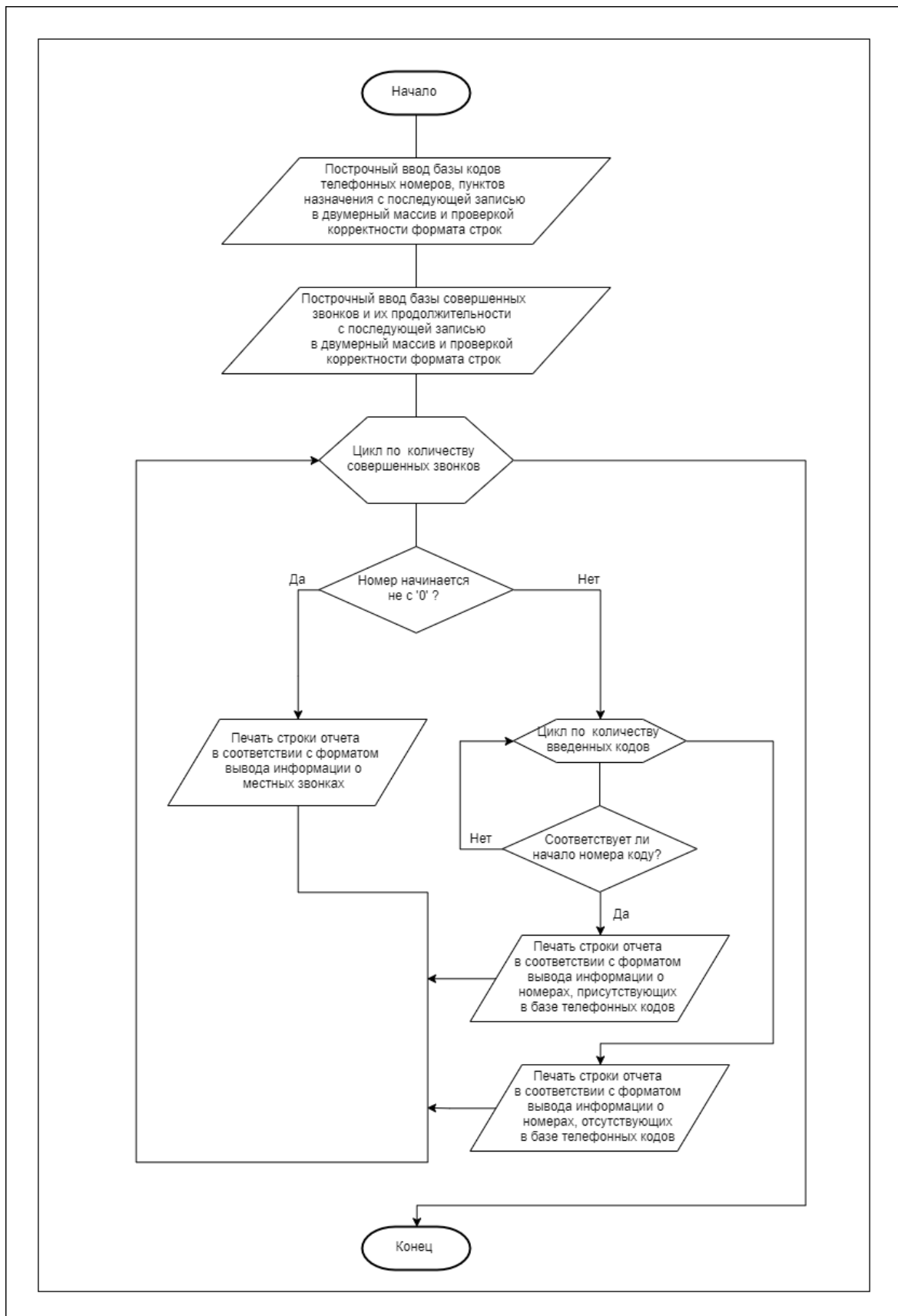
Введение

Цель настоящей работы – разработка программы для решения поставленной задачи, изучение возможностей языка Python и их применение во время создания данного проекта. Для достижения цели необходимо выполнить следующие подзадачи:

1. Проектирование алгоритма.
2. Реализация алгоритма на языке программирования Python.
3. Отладка разработанной программы (поиск на наличие ошибок в ходе работы)
4. Проверка правильности ответов программы после завершения выполнения.
5. Написание документации.

Основная часть

Блок схема



Описание программы

После запуска программы происходит построчный ввод базы данных телефонных кодов. Каждая строка должна соответствовать определенному формату (Код Δ Пунктназначения\$Цена в центах за минуту, где Δ – пробел). Для проверки этого проверяется количество знаков «пробел» и «\$», затем производится еще одна проверка, чтобы до знака «пробел» и после знака «\$» были только цифры, а также между знаком «пробел» и «\$» были только буквы. В случае несоответствия формату пользователь получит сообщение с просьбой повторить ввод строки. В случае корректного ввода строка разделяется по смысловым полям и добавляется в двумерный массив для удобства работы с ней. После ввода строки «000000» завершается заполнение двумерного массива, состоящего из информации из базы телефонных кодов.

Далее происходит ввод информации о телефонных звонках и их продолжительности. Здесь тоже проходит соответствующая проверка корректности ввода (в строке должен быть как минимум один знак «пробел», происходит разделение по последнему знаку «пробел» и проверка, чтобы до были только цифры (разделенные знаком или нет) и после только цифры. В случае несоответствия формату пользователь получит сообщение с просьбой повторить ввод строки. В случае корректного ввода строка разделяется по смысловым полям и добавляется в двумерный массив для удобства работы с ней. После ввода символа «#» завершается заполнение двумерного массива, состоящего из информации из базы совершенных звонков.

Следующим шагом программа запускает цикл по перебору совершенных звонков.

Если первая цифра номера не «0», значит это местный звонок и программа печатает строку отчета в соответствии с форматом вывода информации о местных звонках.

Если же нет, тогда программа запускает цикл по перебору телефонных кодов и ищет совпадение начала телефонного номера и кода.

Если найдено, тогда программа печатает строку в соответствии с форматом вывода информации о номерах, присутствующих в базе телефонных кодов и выходит из цикла поиска соответствия.

Если по завершении цикла совпадения не найдено, тогда программа печатает строку в соответствии с форматом вывода информации о номерах, отсутствующих в базе телефонных кодов.

Листинг

```
#Функция для проверки корректности ввода базы кодов согласно формату
def check_base_line (line):
    if line.count(' ') == 1 and line.count('$') == 1:
        if line[:line.index(' ')].isdigit() \
            and line[line.index(' ')+1:line.index('$')].isalpha() \
            and line[line.index('$')+1:len(line)].isdigit(): return True
        else: return False
    else: return False

#Функция для проверки корректности ввода базы телефонных звонков согласно
формату
def check_calls_line (line):
    if line.count(' ') >= 1 :
        if (line[:line.rindex(' ')].replace(' ','')).isdigit() \
            and line[line.rindex(' ')+1:len(line)].isdigit(): return True
        else: return False
    else: return False

print('*'*80,end = '\n\n')
print('Уважаемый пользователь, Вас приветствует программа для \n\
обработки данных о телефонном звонке и выводе информации \n\
о каждом звонке в формате отчета.', end = '\n\n\n')

#Ввод базы кодов телефонных звонков
print('Вводите международные и междугородние телефонные коды,\n\
пункт назначения и соответствующий тарифный план:')
p_base = []
while True:
    line = input()
    if line == '000000': break

    #Проверка корректности данных
    while not check_base_line(line):
        line = input('Вы ввели неформатную строку, повторите ввод:\n')

    line = line.replace('$',' ')
    line = line.split(' ')
    p_base.append(line)

#Ввод базы телефонных звонков
print('\nВводите номера звонков и их продолжительность:')
p_calls = []
while True:
    line = input()
    if line == '#': break

    #Проверка корректности данных
    while not check_calls_line(line):
        line = input('Вы ввели неформатную строку, повторите ввод:\n')

    line = list(line.rpartition(' '))
    line[0] = line[0].replace(' ','')
    del (line[1])
    p_calls.append(line)
```

```

for i in range(len(p_calls)):
    #Проверка: является ли данный номер местным
    if p_calls[i][0][0]!='0':
        #Печать местных звонков
        print('{:<16}{:<26}{:<15}{:<5}{:<5}{:<5}'.format(p_calls[i][0],\
            'Local', p_calls[i][0], p_calls[i][1], '0.00', '0.00'))
    else:
        found = False
        for j in range(len(p_base)):
            #Проверка: есть ли данный номер в базе телефонных кодов
            if p_calls[i][0].find(p_base[j][0]) == 0:
                found = True
                #Печать звонков, найденных в базе телефонных кодов

print('{:<16}{:<26}{:<15}{:<5}{:<5}{:<5}'.format(p_calls[i][0],\
    p_base[j][1], p_calls[i][0].replace(p_base[j][0], ''),\
    p_calls[i][1], int(p_base[j][2])/100, int(p_base[j][2]) *
int(p_calls[i][1])/100))
        break
        #Печать звонков отсутствующих в базе телефонных кодов
        if found == False:

print('{:<16}{:<26}{:<15}{:<5}{:<5}{:<5}'.format(p_calls[i][0],\
    'Unknown', '', p_calls[i][1], '', '-1.00'))

#Оповещение пользователя об успешном завершении программы
print('\nПрограмма успешно завершена.', end = '\n\n')
print('*'*80)

```


Тесты

Тест 1.

Входные данные:

088925 Broadwood\$81
03 Arrowtown\$38
0061 Australia\$140
000000
031526 22
0061853279 3
08892562287213 122
779760 1
002832769 5
#

Выходные данные:

031526	Arrowtown	1526	22	0.38	8.36
0061853279	Australia	853279	3	1.4	4.2
08892562287213	Broadwood	62287213	122	0.81	98.82
779760	Local	779760	1	0.00	0.00
002832769	Unknown		5		-1.00

Тест 2.

Входные данные:

030489 Москва\$54
04 Чикаго\$21
09 Бостон\$89
0045 China\$140
29 Орел\$12
000000
040489123 43
0045853279 324
005345 5
030489 622 87213 32
29 1 11 1
#

Выходные данные:

040489123	Чикаго	89123	43	0.21	9.03
0045853279	China	853279	324	1.4	453.6
03048962287213	Москва	62287213	32	0.54	17.28
29111	Local	29111	1	0.00	0.00
005345	Unknown		5		-1.00

Тест 3.

Входные данные:

000000
4141 124
01231 1
00234 53

Выходные данные:

4141	Local	4141	124	0.00	0.00
01231	Unknown	1			-1.00
00234	Unknown		53		-1.00

Тест 4.

Входные данные:

030489 Москва\$54
04 Чикаго\$21
09 Бостон\$89
0045 China\$140
29 Орел\$12
000000
#

Выходные данные:

(пусто)

Заключение

В ходе проектной работы я изучил методы работы со строками, функциями и другими возможностями языка Python 3.7.4. Научился решать подобные задачи, разработал алгоритм для реализации поставленной задачи, отладил программу в ходе тестирования и приобрел опыт написания документации.