Федеральное государственное образовательное бюджетное

учреждение высшего образования

«**Финансовый университет**

**при Правительстве Российской Федерации»**

**(Финансовый университет)**

**Департамент анализа данных, принятия решений и финансовых технологий**

**Отчёт по семинару №5**

**Выполнил:**

Студент Малкеров Г.А.

Группы ЗБ-ПИ1-2

**Проверил:**

Петросов Д.А.

**Москва 2020**

**Задания №№1-5:**

1. Создать словарь адресной книги, содержащий ФИО и адрес. Заполнить его 50 элементами, реализовать поиск по адресу

2. Cоздать словарь телефонного справочника. Заполнить его 50 элементами. Реализовать поиск по телефону

3. реализовать проверку на существующие записи в предыдущих заданиях с возможностью дополнения

4. создать словарь на свободную тему, включающий в себя кортеж в качестве ключа, реализовать поиск

5. создать словарь авиарейсов, в возможностью поиска маршрута из точки А в точку В с учётом 1 пересадки

Программный код на языке Python для заданий №1-5:

import traceback

import random

import string

unicNames = [

"Туров Богдан Валериевич",

"Шухевич Виктор Брониславович",

"Майборода Степан Артёмович",

"Чумак Цефас Алексеевич",

"Молчанов Дмитрий Фёдорович",

"Исаков Родион Эдуардович",

"Медведев Конрад Васильевич",

"Юдин Леонард Иванович",

"Таранец Юлиан Романович",

"Капустин Устин Романович",

"Макаров Ян Григорьевич",

"Овчаренко Юрий Вадимович",

"Федункив Ярослав Владимирович",

"Дроздов Шарль Леонидович",

"Коцюбинский Устин Григорьевич",

"Худобяк Геннадий Сергеевич",

"Поляков Йошка Станиславович",

"Миклашевский Цефас Леонидович",

"Кравчук Ждан Сергеевич",

"Моисеенко Любомир Михайлович",

"Павленко Зуфар Львович",

"Миклашевский Чарльз Максимович",

"Лазарев Орест Леонидович",

"Александров Яромир Данилович",

"Михайлов Жерар Григорьевич",

"Киселёв Шамиль Дмитриевич",

"Батейко Емельян Станиславович",

"Васильев Савва Вадимович",

"Многогрешный Чеслав Станиславович",

"Андреев Карен Михайлович",

"Мартынов Клаус Леонидович",

"Захаров Милан Васильевич",

"Котов Ждан Богданович",

"Кондратьев Огюст Дмитриевич",

"Савин Пётр Иванович",

"Зайцев Николай Виталиевич",

"Пономаренко Цицерон Васильевич",

"Павленко Давид Романович",

"Веселов Леон Виталиевич",

"Фомин Виль Борисович",

"Анисимов Роберт Викторович",

"Третьяков Максим Эдуардович",

"Некрасов Шарль Львович",

"Рябов Чеслав Петрович",

"Буров Пётр Валериевич",

"Фомичёв Харитон Богданович",

"Савенко Шамиль Викторович",

"Казаков Георгий Данилович",

"Виноградов Валериан Сергеевич",

"Мухин Устин Васильевич",

]

def PhoneGeneration( n):

return str(random.randint(10 \*\* (n - 1), (10 \*\* n) - 1))

def Task1DicGenerator():

n = unicNames

d = {}

for i in range(0, len(n), 2):

d["адрес №" + str(i)] = [n[i], n[i + 1]]

print(f"Итого для поиска\n{d}")

return d

def Task2DicGenerator():

d = {}

for e in unicNames:

key = "+7" + PhoneGeneration(10)

d[key] = e

print(f"Итого для поиска\n{d}")

return d

def Task1ToThird():

d=Task1DicGenerator()

intputInfo = input("Введите адрес для поиска:")

if intputInfo in d:

print("Адрес найден!\nПроживающие по адресу:")

[print(x) for x in d[intputInfo]]

else:

print("Адрес не найден, но мы его добавим в систему")

d[intputInfo] = input("Введите ФИО людей, проживающих по этому адресу через точку с запятой: ").split(",")

print("Обновлённый словарь:")

for k, v in d.items():

print(k, v)

def Task2ToThird():

d=Task2DicGenerator()

inputInfo = input("Введите номер телефона для поиска:")

if inputInfo in d:

print("Телефон найден!\nАбонент " + d[inputInfo])

else:

print("Абонент не найден, но мы его можем добавить")

name = input("Введите ФИО абонента ")

d[inputInfo] = name

print("Обновлённый словарь:")

for k, v in d.items():

print(k, v)

def Task1():

d = Task1DicGenerator()

s = input("Введите адрес для поиска:")

if s in d:

print("Адрес найден!\nПроживающие по адресу:")

[print(x) for x in d[s]]

else:

print("Данные не найдены!")

def Task2():

d =Task2DicGenerator()

inputInfo = input("Введите номер телефона для поиска:")

if inputInfo in d:

print("Значение для телефона " + inputInfo + " : " + d[inputInfo])

else:

print("Введенные данные НЕ были найдены")

def Task3():

tasks = {

1: Task1ToThird,

2: Task2ToThird,

}

numTask=int (input("Введите номер задания для работы: "))

if(numTask in tasks):

tasks[numTask]()

else:

print("Введенного номера нет в задании")

def allCounter(\*t):

res = 0

for i in t:

res += i

return res

def allMultipy( \*t):

res = 1

for i in t:

res \*= i

return res

def checkDigit( e):

try:

return int(e)

except:

return e

def Task4():

max\_values = 100

d={}

for i in range(max\_values):

for j in range(max\_values):

for k in range(max\_values):

d[(i, j, k)] = [{"multiplication": allMultipy(i, j, k), "sum": allCounter(i, j, k)}]

r = d[tuple([checkDigit(x) for x in input("Введите 3 числа через пробел от 0 до 100 для быстрого подсчёта их суммы и произведения: ").split(" ")])][0]

print("Произведение: " + str(r["multiplication"]) + "\nСумма: " + str(r["sum"]))

def GetRandomFlight():

letters = string.ascii\_uppercase

chars = ''.join(random.choice(letters) for i in range(3))

numbers = str(random.randint(1000, 9999))

return chars + "-" + numbers

def Task5():

d = {

"Москва": ["Лондон", "Владивосток", "Санкт-Петербург"],

"Лондон": ["Москва", "Сингапур"],

"Сингапур": ["Лондон"],

"Калининград": ["Санкт-Петербург"],

"Санкт-Петербург": ["Калининград", "Москва"],

"Владивосток": ["Москва", "Норильск"],

"Норильск": ["Владивосток"],

}

flightInfo = {}

for e in d:

for city in d[e]:

flightInfo[GetRandomFlight()] = {"from": e, "to": city}

searchFlag = False

pointA = input("Введите точку А: ")

pointB = input("Введите точку В: ")

beginingFlag = False

for flight, value in flightInfo.items():

if value["from"] == pointA:

beginingFlag = True

if beginingFlag == False:

print("Нет указанной точки вылета в словаре!")

return

for flight, value in flightInfo.items():

if value["from"] == pointA and value["to"] == pointB:

searchFlag = True

print("План перелёта\nСуществует прямой рейс №" + flight + "\n" + pointA + " -> " + pointB)

if searchFlag == False:

bufList = []

for flight, value in flightInfo.items():

if value["from"] == pointA:

bufList.append((flight, value["from"], value["to"]))

for element in bufList:

for flight, value in flightInfo.items():

if element[2] == value["from"] and pointB == value["to"]:

searchFlag = True

print("План перелёта:\nНа рейсе №" + element[0] + " " + element[1] + " -> " + element[2])

print("На рейсе №" + flight + " " + value["from"] + " -> " + value["to"])

if searchFlag == False:

print("По вашему запросу ничего не найдено")

def main():

try:

print("Начало работы программы")

tasks = {

1: Task1,

2: Task2,

3: Task3,

4: Task4,

5: Task5,

6: Task6,

}

taskForWork = int(input("Введите номер задания от 1 до 6: "))

if(taskForWork in tasks):

tasks[taskForWork]()

else:

print("Задание не найдено")

print("Конец работы программы")

except Exception:

print("Ошибка при работе программы ", traceback.format\_exc())

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Пример работы программного кода:**

Пример работы программного кода из заданий №№1-5 представлен на рисунке 1-6.

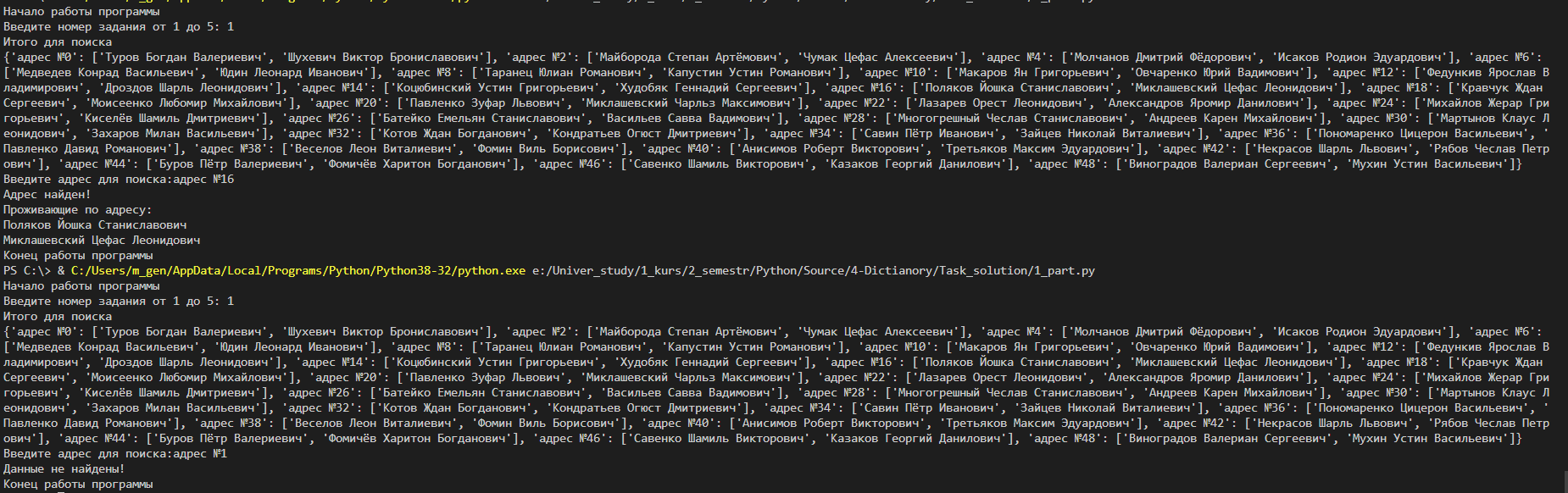


Рисунок 1 – Пример работы программного кода из задания №1

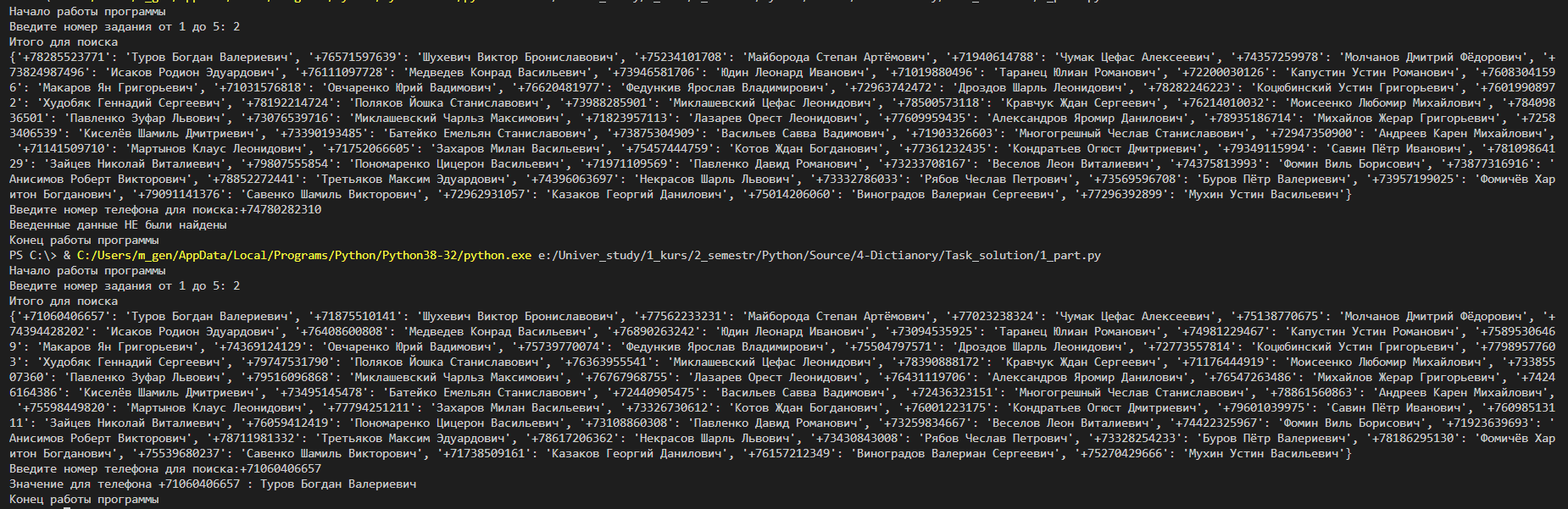


Рисунок 2 – Пример работы программного кода из задания №2

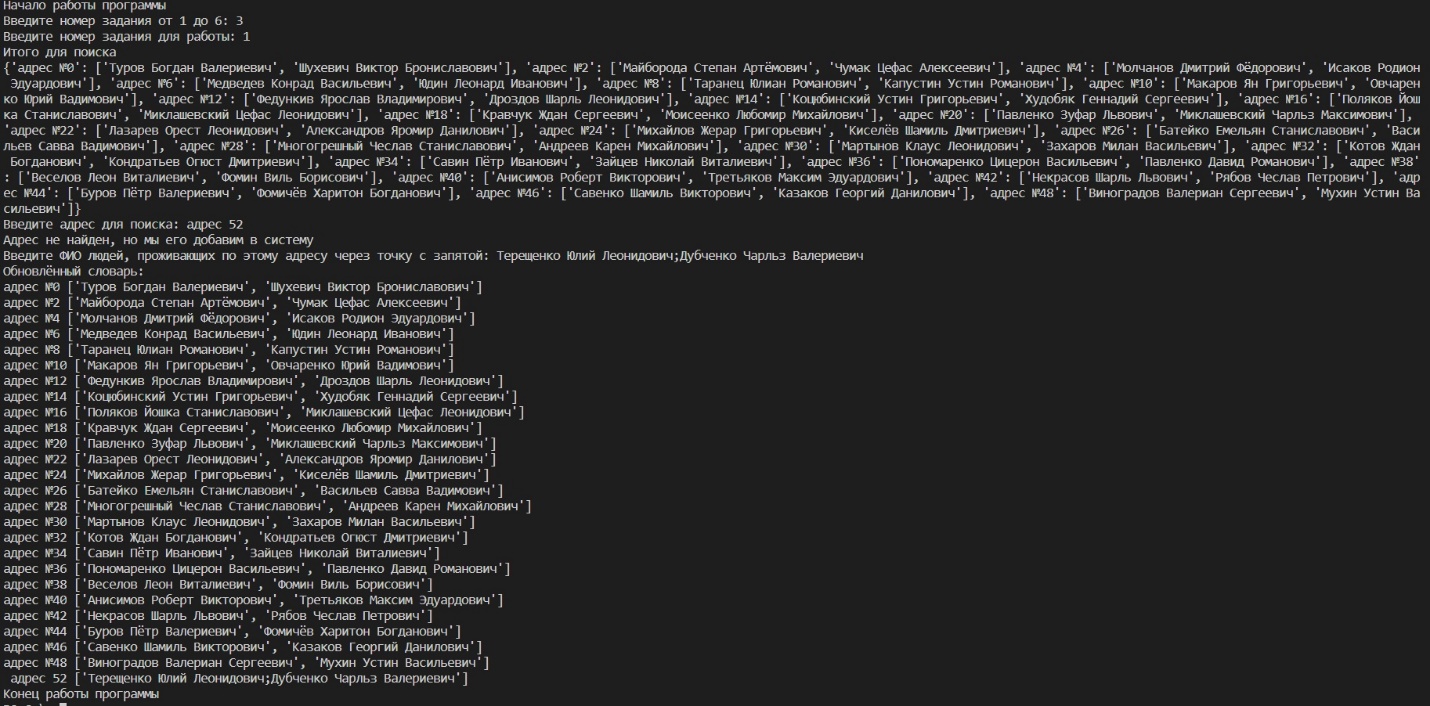


Рисунок 3 – Пример работы программного кода из задания №3 часть a

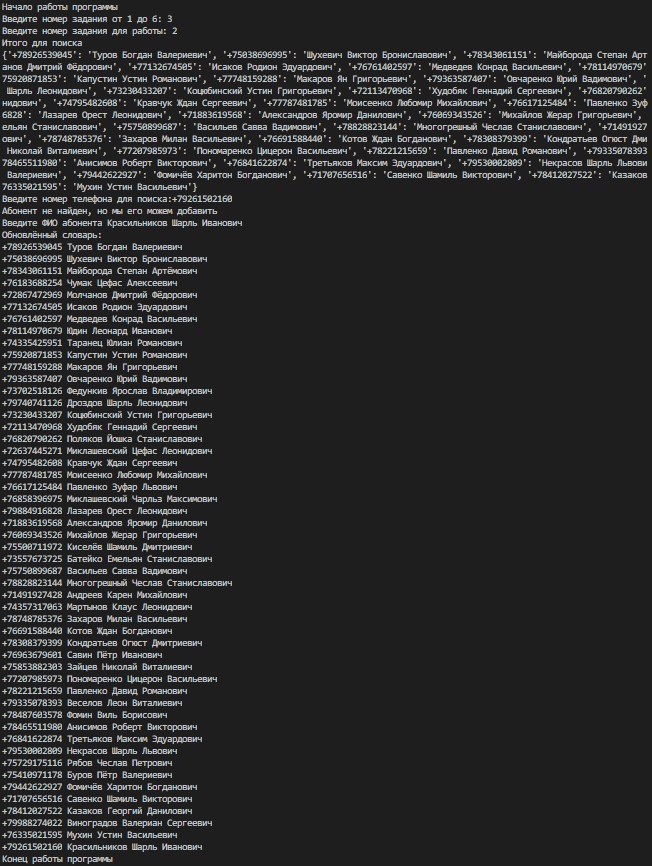


Рисунок 4 – Пример работы программного кода из задания №3 часть b

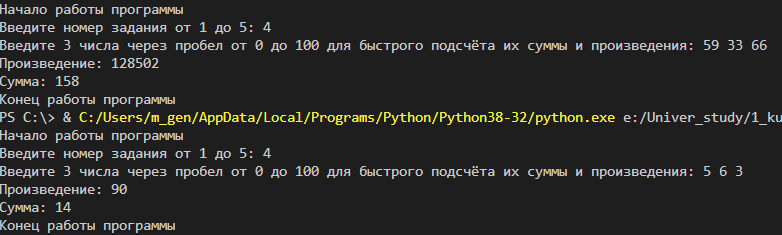


Рисунок 5 – Пример работы программного кода из задания №4

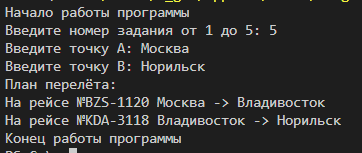


Рисунок 6 – Пример работы программного кода из задания №5

**Вывод:** В ходе выполнения данной практической работы были получены навыки работы со словарями в языке программирования Python