Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №9
дисциплины «Программирование на Python»
Вариант 9

Выполнил:
Дудкин Константин Александрович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»
направление «Программное
обеспечение средств вычислительной
техники и автоматизированных
систем»,
очная форма обучения

_____
(подпись)

Руководитель практики:
Кандидат технических наук, доцент
кафедры инфокомуникаций, доцент
Воронкин Роман Александрович

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

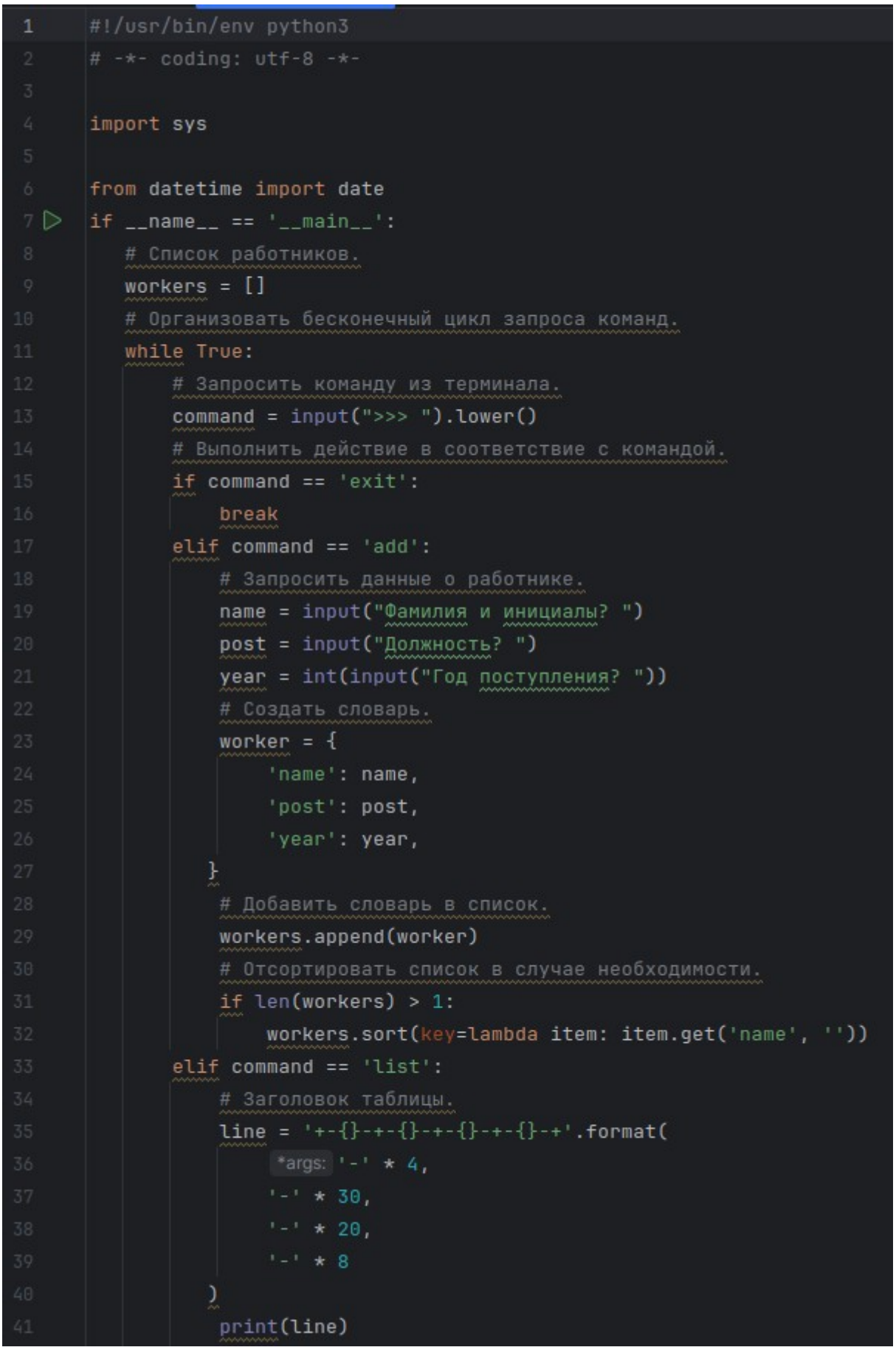
Тема: Работа со словарями в языке Python

Цель: Приобретение навыков по работе со словарями при написании программ с помощью языка программирования Python версии 3.x

Порядок выполнения работы

1. Проработал пример: Использовать словарь, содержащий следующие ключи: фамилия и инициалы работника; название заниаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в список, состоящий из заданных словарей
- записи должны быть размешены по алфавиту
- вывод на дисплей фамилии работников, чей стаж работы в организации превышает значение, введенное с клавиатуры
- если таких работников нет, вывести на дисплей соответствующее сообщение



```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

from datetime import date
if __name__ == '__main__':
    # Список работников.
    workers = []
    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()
        # Выполнить действие в соответствие с командой.
        if command == 'exit':
            break
        elif command == 'add':
            # Запросить данные о работнике.
            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))
            # Создать словарь.
            worker = {
                'name': name,
                'post': post,
                'year': year,
            }
            # Добавить словарь в список.
            workers.append(worker)
            # Отсортировать список в случае необходимости.
            if len(workers) > 1:
                workers.sort(key=lambda item: item.get('name', ''))
        elif command == 'list':
            # Заголовок таблицы.
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                *args: '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 8
            )
            print(line)
```

Рисунок 1. Код программы примера (часть 1)

```
            print(
                '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
                    *args: "№",
                    "Ф.И.О.",
                    "Должность",
                    "Год"
                )
            )
            print(line)
            # Вывести данные о всех сотрудниках.
            for idx, worker in enumerate(workers, 1):
                print(
                    '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
                        *args: idx,
                        worker.get('name', ''),
                        worker.get('post', ''),
                        worker.get('year', 0)
                    )
                )
            print(line)
        elif command.startswith('select '):
            # Получить текущую дату.
            today = date.today()
            # Разбить команду на части для выделения номера года.
            parts = command.split( sep: ' ', maxsplit=1)
            # Получить требуемый стаж.
            period = int(parts[1])
            # Инициализировать счетчик.
            count = 0
            # Проверить сведения работников из списка.
            for worker in workers:
                if today.year - worker.get('year', today.year) >= period:
                    count += 1
                    print(
                        '{:>4}: {}'.format( *args: count, worker.get('name',
                    )
            # Если счетчик равен 0, то работники не найдены.
            if count == 0:
                print("Работники с заданным стажем не найдены.")
```

Рисунок 2. Код программы примера (часть 2)

```
            print("Работники с заданным стажем не найдены.")
        elif command == 'help':
            # Вывести справку о работе с программой.
            print("Список команд:\n")
            print("add - добавить работника;")
            print("list - вывести список работников;")
            print("select <стаж> - запросить работников со стажем;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)
```

Рисунок 3. Код программы примера (часть 3)

```
/usr/bin/python3.11 /home/code_ra1der/git/Python_LW9/Example.py
>>> add
Фамилия и инициалы? Дудкин Константин
Должность? Учащийся
Год поступления? 2022
>>>
>>> Неизвестная команда

Неизвестная команда
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |             Ф.И.О.               |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Дудкин Константин                | Учащийся             |     2022 |
+------+----------------------------------+----------------------+----------+
>>> exit

Process finished with exit code 0
```

Рисунок 4. Результат программы

2. Выполнил основное задание №1: Создайте словарь, связав его с переменной school, и наполните данными, которые бы отражали количество учащихся в разных классах (1а, 1б, 2б, 6а, 7в и т.п.). Внесите изменения в словарь согласно следующему: а) в одном из классов изменилось количество учащихся, б) в школе появился новый класс, с) в школе был расформирован (удален) другой класс. Вычислите общее количество учащихся в школе

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':

    school = {
        '1а': 17,
        '1б': 30,
        '2б': 28,
        '6а': 19,
        '7в': 15,
    }

    print("Исходный словарь:")
    print(school)

    total_students = sum(school.values())
    print("\nОбщее количество учащихся в школе:", total_students)
    # Задача а)
    school['1а'] = 21
    # Задача б)
    school['8г'] = 26
    # Задача с)
    if '2б' in school:
        del school['2б']

    print("\nИзмененный словарь:")
    print(school)

    total_students = sum(school.values())
    print("\nОбщее количество учащихся в школе:", total_students)
```

Рисунок 5. Код программы основного задания 1

Рисунок 6. Результат работы программы

```
/usr/bin/python3.11 /home/code_ra1der/git/Python_LW9/Task1.py
Исходный словарь:
{'1а': 17, '1б': 30, '2б': 28, '6а': 19, '7в': 15}

Общее количество учащихся в школе: 109

Измененный словарь:
{'1а': 21, '1б': 30, '6а': 19, '7в': 15, '8г': 26}

Общее количество учащихся в школе: 111

Process finished with exit code 0
```

3. Выполнил основное задание 2: Создайте словарь, где ключами являются числа, а значениями — строки. Примените к нему метод items(), с помощью полученного объекта dict_items слздайте новый словарь, «обратный» исходному, т.е. ключами являются строки, а значениями — числа

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':

    original_dict = {
        1: 'один',
        2: 'два',
        3: 'три',
        4: 'четыре',
    }

    dict_items_object = original_dict.items()

    reversed_dict = {value: key for key, value in dict_items_object}

    print("Исходный словарь:")
    print(original_dict)

    print("\nНовый словарь (обратный):")
    print(reversed_dict)
```

Рисунок 7. Код программы основного задания 2

Рисунок 8. Результат работы программы

4. Выполнил индивидуальное задание: Использовать словарь, содержащий следующие ключи: название начального пункта маршрута, номер маршрута. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры, записи должны быть упорядочены по номерам маршрутов, вывод на экран информации о маршруте, номер которого введен с клавиатуры. Если таких маршрутов нет, выдать на дисплей соответствующее сообщение

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    # Список маршрутов
    routes = []
    number = 0
    # Начало бесконечного цикла команд
    while True:
        # Сюда вписывать команды
        command = input('>>> ').lower()

        # Команда help
        if command == 'help':
            print('\nСписок команд:')
            print('help - Вывести этот список')
            print('add - Добавить маршрут')
            print('list - Показать список маршрутов')
            print('exit - Выйти из программы')
        # Команда add
        elif command == 'add':
            # Запись данных маршрута
            first = input('Первая точка маршрута: ')
            second = input('Вторая точка маршрута: ')
            number += 1
            # Создание словаря
            route = {
                'first_checkpoint': first,
                'second_checkpoint': second,
                'number': number
            }

            # Добавление словаря в список
            routes.append(route)
            # Сортировка по номеру маршрута
            routes.sort(key=lambda item: item.get('number', ''))
```

Рисунок 9. Код программы задания (часть 1)

```
        # Команда list
        elif command == 'list':
            # Заголовок таблицы.
            line = '+-{}-+-{}-+-{}-+'.format(
                *args: '-' * 14,
                '-' * 20,
                '-' * 20
            )
            print(line)
            print(
                '| {:^5} | {:^20} | {:^20} |'.format(
                    *args: "Номер маршрута",
                    "Место отправки",
                    "Место прибытия"
                )
            )
            print(line)
            # Вывод данных о маршрутах
            for route in routes:
                print(
                    '| {:<14} | {:<20} | {:<20} |'.format(
                        *args: route.get('number', ''),
                        route.get('first_checkpoint', ''),
                        route.get('second_checkpoint', '')
                    )
                )
            print(line)

        # Команда exit
        elif command == 'exit':
            break
        # Другая команда/неверно введенная команда
        else:
            print(f'Неизвестная команда {command}')
```

Рисунок 10. Код программы задания (часть 2)

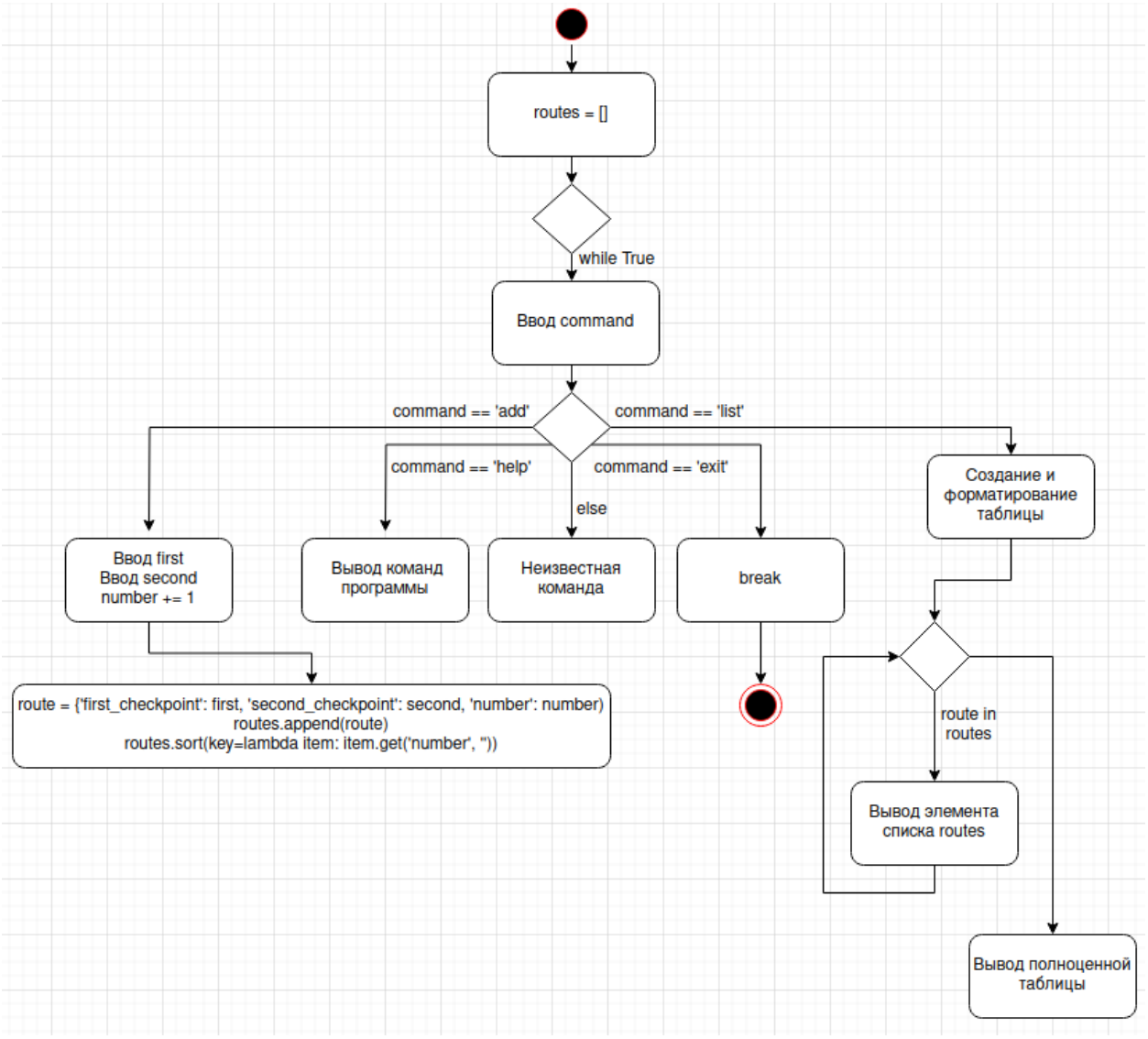


Рисунок 11. UML-диаграмма программы

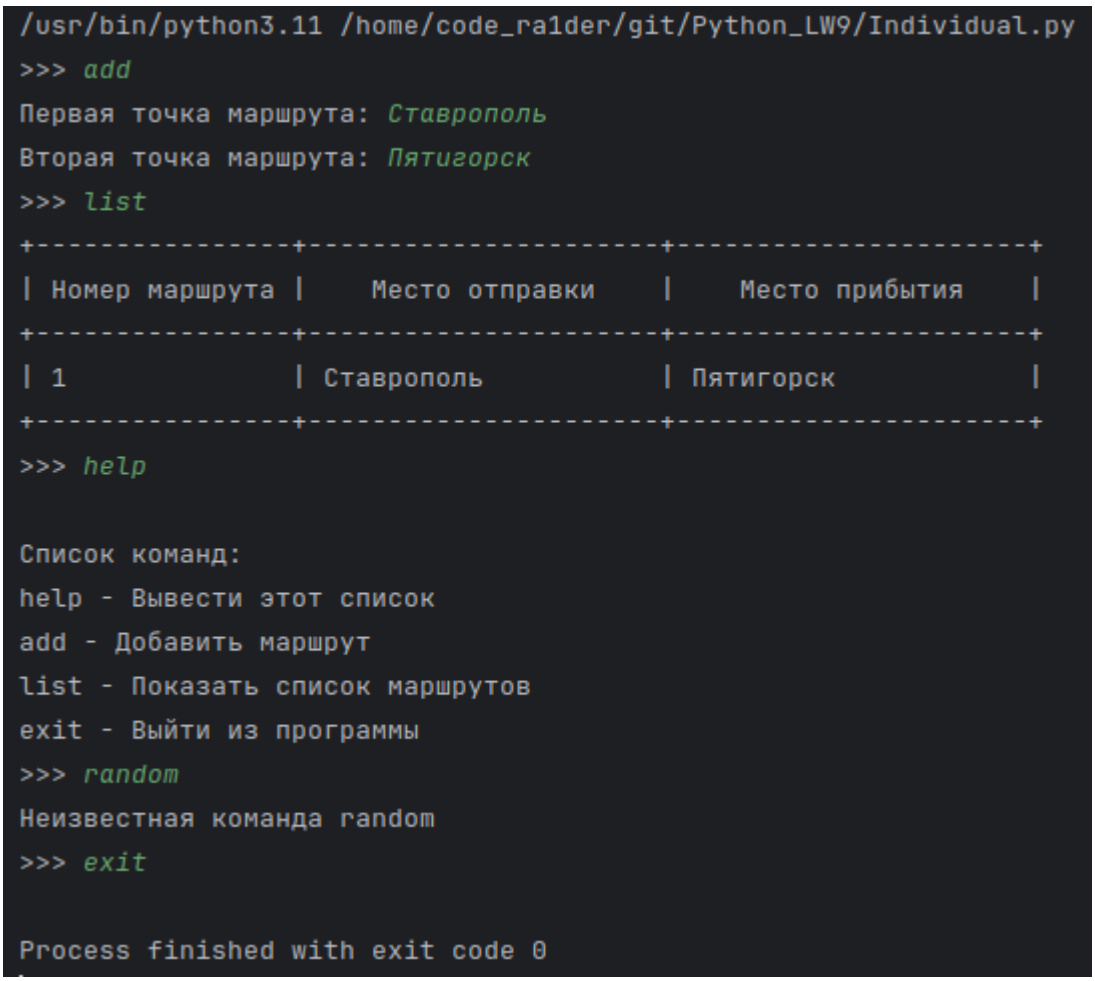


Рисунок 12. Результат работы программы

Ответы на вопросы

1. Словарь представляет собой структуру данных, предназначенную для хранения произвольных объектов с доступом по ключу

2. Да, может, с выводом количества ключей в данном словаре

3. Цикл for, методы items(), keys() и values()

4. Для доступа значения из словаря по ключу можно воспользоваться методом get() или записью ключа в квадратные скобки

5. С помощью setdefault() можно добавить элемент в словарь. С помощью квадратных скобок: duct[„a“] = 1. С помощью метода update(): dict.update({„a“:1})

6. Словарь включений — схожая со списковыми включениями структура, за исключением лишь того, что он создает объект словаря вместо списка

7. Функция zip() принимает итерируемый объект (например, список, кортеж, множество или словарь) как аргумент. Затем она генерирует список кортежей, которые содержат элементы из каждого объекта, переданного в функцию

Пример: keys = [„a“, „b“, „c“]

values = [1, 2, 3]

dict = dict(zip(keys, values))

Вывод: {„a“: 1, „b“: 2, „c“: 3}

8. Модуль datetime в Python позволяет работать с датой и временем, предоставляя классы и функции для работы с текущей датой и временм, возможность их форматирования, а также вычисления разницы межу двумя датами посредством класса timedelta

Вывод: В ходе выполнения работы были изучены словари и приобретены навыки по работе с ними посредством программирования на языке программирования Python версии 3.x