

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Работа со словарями в языке Python»**

**ОТЧЕТ**  
**по лабораторной работе №9**  
**дисциплины**  
**«Основы программной инженерии»**

Выполнил:

Гълбачева Доротея Андреева  
2 курс, группа ПИЖ-б-о-21-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

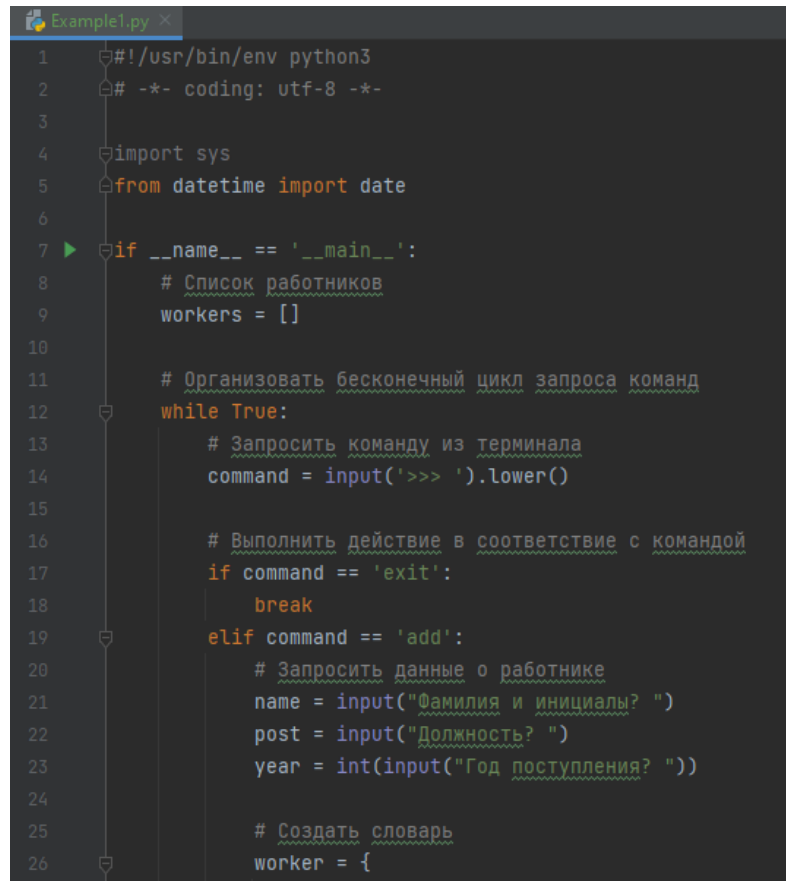
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

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

### **Выполнения лабораторной работы:**

#### **1. Проработка примеров из лабораторной работы:**



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5  from datetime import date
6
7  if __name__ == '__main__':
8      # Список работников
9      workers = []
10
11     # Организовать бесконечный цикл запроса команд
12     while True:
13         # Запросить команду из терминала
14         command = input('>>> ').lower()
15
16         # Выполнить действие в соответствие с командой
17         if command == 'exit':
18             break
19         elif command == 'add':
20             # Запросить данные о работнике
21             name = input("Фамилия и инициалы? ")
22             post = input("Должность? ")
23             year = int(input("Год поступления? "))
24
25             # Создать словарь
26             worker = {
```

Рисунок 9.1 - Код программы примера №1

```

25         # Создать словарь
26         worker = {
27             'name': name,
28             'post': post,
29             'year': year,
30         }
31
32         # Добавить словарь в список
33         workers.append(worker)
34         # Отсортировать список в случае необходимости
35         if len(workers) > 1:
36             workers.sort(key=lambda item: item.get('name', ''))
37
38     elif command == 'list':
39         # Заголовок таблицы
40         line = '+--{}--{}--{}--{}--+'.format(
41             '-' * 4,
42             '-' * 30,
43             '-' * 20,
44             '-' * 8
45         )
46         print(line)
47         print(
48             '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
49                 "№",
50                 "Ф.И.О.",

```

Рисунок 9.2 - Код программы примера №1

```

Example1.py x
47         print(
48             '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
49                 "№",
50                 "Ф.И.О.",
51                 "Должность",
52                 "Год"
53             )
54         )
55         print(line)
56
57         # Вывести данные о всех сотрудниках
58         for idx, worker in enumerate(workers, 1):
59             print(
60                 '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
61                     idx,
62                     worker.get('name', ''),
63                     worker.get('post', ''),
64                     worker.get('year', 0)
65                 )
66             )
67
68         print(line)
69
70     elif command.startswith('select'):
71         # Получить текущую дату
72         today = date.today()

```

Рисунок 9.3 - Код программы примера №1

```

70 elif command.startswith('select'):
71     # Получить текущую дату
72     today = date.today()
73
74     # Разбить команду на части для выделения номера года
75     parts = command.split(' ', maxsplit=1)
76     # Получить требуемый стаж
77     period = int(parts[1])
78
79     # Инициализировать счетчик
80     count = 0
81     # Проверить сведения работников из списка
82     for worker in workers:
83         if today.year - worker.get('year', today.year) >= period:
84             count += 1
85             print(
86                 '{:>4}: {}'.format(count, worker.get('name', ''))
87             )
88     # Если счетчик равен 0, то работники не найдены
89     if count == 0:
90         print("Работники с заданным стажем не найдены.")
91
92 elif command == 'help':
93     # Вывести справку о работе с программой
94     print("Список команд:\n")
95     print("add - добавить работника;")

```

Рисунок 9.4 - Код программы примера №1

```

79     # Инициализировать счетчик
80     count = 0
81     # Проверить сведения работников из списка
82     for worker in workers:
83         if today.year - worker.get('year', today.year) >= period:
84             count += 1
85             print(
86                 '{:>4}: {}'.format(count, worker.get('name', ''))
87             )
88     # Если счетчик равен 0, то работники не найдены
89     if count == 0:
90         print("Работники с заданным стажем не найдены.")
91
92 elif command == 'help':
93     # Вывести справку о работе с программой
94     print("Список команд:\n")
95     print("add - добавить работника;")
96     print("list - вывести список работников;")
97     print("select <стаж> - запросить работников со стажем;")
98     print("help - отобразить справку;")
99     print("exit - завершить работу с программой.")
100
101 else:
102     print(f"Неизвестная команда {command}, file=sys.stderr")

```

Рисунок 9.5 - Код программы примера №1

```
Example1 x
C:\Users\user\LabR_9\venv\Scripts\python.exe C:\Users\user\LabR_9\Example1.py
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Фамилия и инициалы? Иванов А.Г.
Должность? Геймдизайнер
Год поступления? 2017
>>> add
Фамилия и инициалы? Петров Д.К.
Должность? Программист
Год поступления? 2021
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Иванов А.Г. | Геймдизайнер | 2017 |
| 2 | Петров Д.К. | Программист | 2021 |
+-----+-----+-----+-----+
>>> select 5
1: Иванов А.Г.
>>> select 10
Работники с заданным стажем не найдены.
>>> exit
```

Рисунок 9.6 - Результат работы программы примера №1

## 2. Индивидуальные задания:

### Задание №1:

- Использовать словарь, содержащий следующие ключи: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по номерам поездов; вывод на экран информации о поезде, номер которого введен с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.

```

Example1.py x Individual1.py x
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5  from datetime import time
6
7  if __name__ == '__main__':
8      trains = []
9      while True:
10         command = input('>>> ').lower()
11
12         if command == 'exit':
13             break
14
15         elif command == 'add':
16             # Запросить данные о работнике
17             destination = input("Название пункта назначения: ")
18             number = int(input("Номер поезда: "))
19             time = input("Время отправления: ")
20
21             # Создать словарь
22             train = {
23                 'destination': destination,
24                 'number': number,
25                 'time': time,
26

```

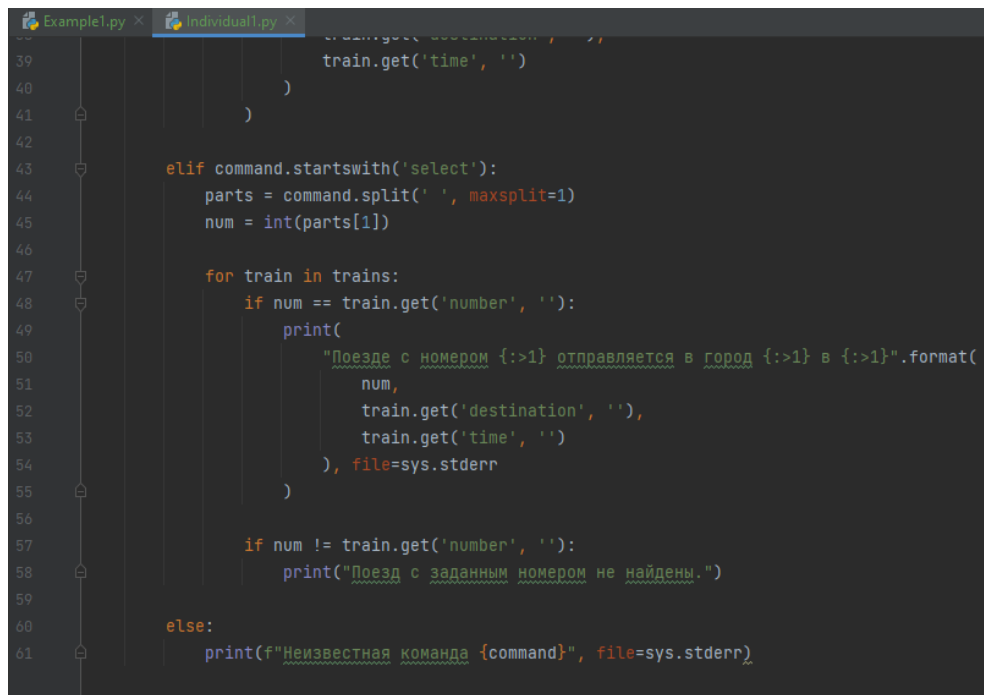
Рисунок 9.7 - Код программы задания №1

```

Example1.py x Individual1.py x
26         }
27
28         trains.append(train)
29         if len(trains) > 1:
30             trains.sort(key=lambda item: item.get('number', ''))
31
32         elif command == 'list':
33             for idx, train in enumerate(trains, 1):
34                 print(
35                     '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
36                         idx,
37                         train.get('number', 0),
38                         train.get('destination', ''),
39                         train.get('time', '')
40                     )
41                 )
42
43         elif command.startswith('select'):
44             parts = command.split(' ', maxsplit=1)
45             num = int(parts[1])
46
47             for train in trains:
48                 if num == train.get('number', ''):
49                     print(
50                         "Поезде с номером {:>1} отправляется в город {:>1} в {:>1}".format(
51                             num,

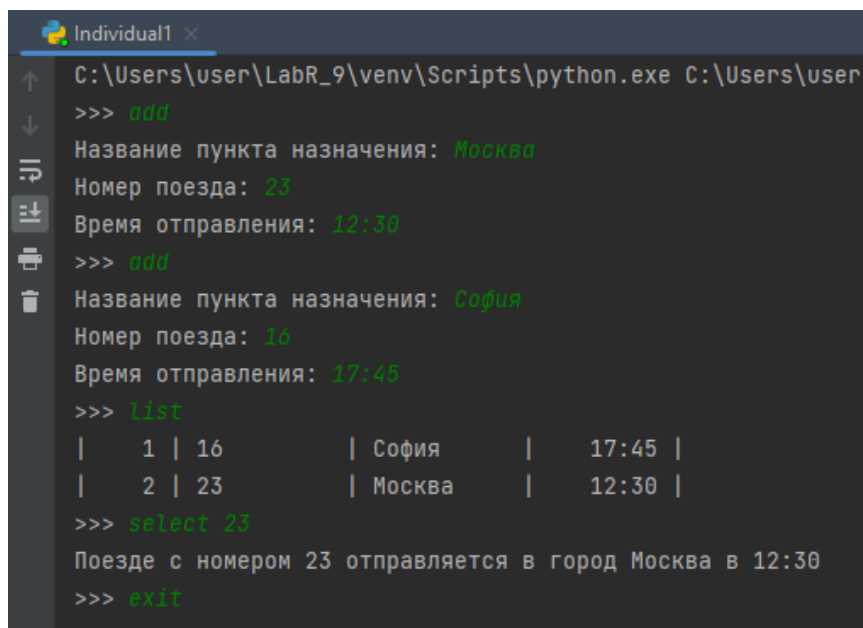
```

Рисунок 9.8 - Код программы задания №1



```
39         train.get('destination', '')
40     )
41 )
42
43 elif command.startswith('select'):
44     parts = command.split(' ', maxsplit=1)
45     num = int(parts[1])
46
47     for train in trains:
48         if num == train.get('number', ''):
49             print(
50                 "Поезде с номером {:>1} отправляется в город {:>1} в {:>1}".format(
51                     num,
52                     train.get('destination', ''),
53                     train.get('time', '')
54                 ), file=sys.stderr
55             )
56
57         if num != train.get('number', ''):
58             print("Поезд с заданным номером не найден.")
59
60 else:
61     print(f"Неизвестная команда {command}", file=sys.stderr)
```

Рисунок 9.9 - Код программы задания №1



```
Individual1 x
C:\Users\user\LabR_9\venv\Scripts\python.exe C:\Users\user
>>> add
Название пункта назначения: Москва
Номер поезда: 23
Время отправления: 12:30
>>> add
Название пункта назначения: София
Номер поезда: 16
Время отправления: 17:45
>>> list
| 1 | 16 | София | 17:45 |
| 2 | 23 | Москва | 12:30 |
>>> select 23
Поезде с номером 23 отправляется в город Москва в 12:30
>>> exit
```

Рисунок 9.10 - Результат работы программы задания №1

### 3. Контрольные вопросы:

#### 1. Что такое словари в языке Python?

Словарь (dict) представляет собой структуру данных (которая ещё называется ассоциативный массив), предназначенную для хранения

произвольных объектов с доступом по ключу. Данные в словаре хранятся в формате ключ – значение.

## 2. Может ли функция len() быть использована при работе со словарями?

Функция len() широко используется для определения размера объектов в Python. В нашем случае передача объекта словаря этой функции вернет размер словаря, то есть количество пар ключ-значение, присутствующих в словаре.

## 3. Какие методы обхода словарей Вам известны?

Элементы словаря перебираются в цикле for также, как элементы других сложных объектов. Однако "по-умолчанию" извлекаются только ключи.

С другой стороны у словаря как класса есть метод items(), который создает особую структуру, состоящую из кортежей. Каждый кортеж включает ключ и значение.

Методы словаря keys() и values() позволяют получить отдельно перечни ключей и значений. Так что если, например, надо перебрать только значения или только ключи, лучше воспользоваться одним из этих методов.

## 4. Какими способами можно получить значения из словаря по ключу?

```
>>> for i in nums:
```

```
... print(nums[i])
```

```
...
```

```
one
```



two

three

5. Какими способами можно установить значение в словаре по ключу?

С помощью метода `setdefault()`, при непосредственном обращении к ключу словарю.

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.

```
>>> {x: x * x for x in (1, 2, 3, 4)}  
{1: 1, 2: 4, 3: 9, 4: 16}
```

7. Самостоятельно изучите возможности функции `zip()` приведите примеры ее использования.

Функция `zip()` в Python создает итератор, который объединяет элементы из нескольких источников данных. Эта функция работает со списками, кортежами, множествами и словарями для создания списков или кортежей, включающих все эти данные. У функции `zip()` множество сценариев применения. Например, она пригодится, если нужно создать набор словарей из двух массивов, каждый из которых содержит имя и номер сотрудника. Функция `zip()` принимает итерируемый объект, например, список, кортеж, множество или словарь в качестве аргумента. Затем она генерирует список кортежей, которые содержат элементы из каждого объекта, переданного в функцию. Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию `zip()`.

**8.** Самостоятельно изучите возможности модуля `datetime`. Каким функционалом по работе с датой и временем обладает этот модуль?

Модуль `datetime` предоставляет классы для обработки времени и даты разными способами. Поддерживается и стандартный способ представления времени, однако больший упор сделан на простоту манипулирования датой, временем и их частями.