

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 5.**

Выполнила:  
Михеева Елена Александровна  
2 курс, группа ИВТ-б-з-20-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных  
систем», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А....

---

(подпись)

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

Ставрополь, 2023 г.

## Тема: Работа со словарями в языке Python

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

### Порядок выполнения работы.

1. Были проработаны примеры лабораторной работы. В отчете представлен скриншот работы программы примера 1.

```
>>> add
Фамилия и инициалы? Михеева Е.А.
Должность? Директор
Год поступления? 2001
>>> 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? Козлов В.И.
Должность? Уборщик
Год поступления? 2015
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          | Должность | Год |
+-----+-----+-----+-----+
| 1  | Козлов В.И.              | Уборщик   | 2015 |
| 2  | Михеева Е.А.             | Директор  | 2001 |
+-----+-----+-----+-----+
>>> select 8
1: Козлов В.И.
2: Михеева Е.А.
>>> exit
(base) mikheeva@MacBook-Pro-Elena Python 2.6 %
```

Рисунок 1. Выполнение программы примера 1

3. Было выполнено задание №1.

```
4 # Создайте словарь, связав его с переменной school,
5 # и наполните данными, которые бы отражали количество учащихся
6 # в разных классах (1а, 1б, 2б, 6а, 7в и т. п.).
7 # Внесите изменения в словарь согласно следующему:
8 # а) в одном из классов изменилось количество учащихся,
9 # б) в школе появился новый класс,
10 # в) в школе был расформирован (удален) другой класс.
11 # Вычислите общее количество учащихся в школе
12
13
14 if __name__ == '__main__':
15     school = {
16         '1a': 30,
17         '1б': 30,
18         '1в': 28,
19         '2a': 28,
20         '2б': 26,
21         '3a': 26,
22         '3б': 31,
23     }
24     print("School: ")
25     print(school)
26
27     # а) в одном из классов изменилось количество учащихся
28     change_class = input("Enter the class name... ")
29     change_num = int(input("Enter new number of student... "))
30
31     school[change_class] = change_num
32     print("School: ")
33     print(school)
34
35     # б) в школе появился новый класс
36     new_class = input("Enter the new class name... ")
37     new_num = int(input("Enter the number of student... "))
38
39     school[new_class] = new_num
40     print("School: ")
41     print(school)
42
43     # в) в школе был расформирован (удален) другой класс
44     del_class = input("Enter the name class to delete it: ")
45     del school[del_class]
46     print("School: ")
47     print(school)
48
49     total = sum(school.values())
50     print(f"The total number of student is {total}.")
51
```

Рисунок 2. Программа для задания №1

```
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ

School:
{'1a': 30, '1б': 30, '1в': 28, '2a': 28, '2б': 26, '3a': 26, '3б': 31}
Enter the class name... 1б
Enter new number of student... 26
School:
{'1a': 30, '1б': 30, '1в': 26, '2a': 28, '2б': 26, '3a': 26, '3б': 31}
Enter the new class name... 11a
Enter the number of student... 20
School:
{'1a': 30, '1б': 30, '1в': 26, '2a': 28, '2б': 26, '3a': 26, '3б': 31, '11a': 20}
Enter the name class to delete it: 11a
School:
{'1a': 30, '1б': 30, '1в': 26, '2a': 28, '2б': 26, '3a': 26, '3б': 31}
The tottal number of student is 197.
(base) mikheeva@MacBook-Pro-Elena Python_2.6 %
```

Рисунок 3. Результаты работы программы задания №1

#### 4. Было выполнено задание №2.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # создайте словарь, где ключами являются числа, а значениями — строки.
5  # Примените к нему метод items(), с с помощью полученного объекта
6  # dict_items создайте новый словарь, "обратный" исходному,
7  # т. е. ключами являются строки, а значениями — числа.
8
9  if __name__ == '__main__':
10     original_dict = {
11         1: 'one',
12         2: 'two',
13         3: 'three',
14         4: 'four',
15         5: 'five',
16         6: 'six',
17         7: 'seven',
18         8: 'eight',
19         9: 'nine',
20         10: 'ten'
21     }
22
23     print("Original dictionary: ")
24     print(original_dict)
25
26     swapped = {v: k for k, v in original_dict.items()}
27
28     print("\nReversed dictionary: ")
29     print(swapped)
30
ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ    ПОРТЫ    + ~ Python Debu

Original dictionary:
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 6: 'six', 7: 'seven', 8: 'eight', 9: 'nine', 10: 'ten'}

Reversed dictionary:
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5, 'six': 6, 'seven': 7, 'eight': 8, 'nine': 9, 'ten': 10}
(base) mikheeva@MacBook-Pro-Elena Python_2.6 %
```

Рисунок 4. Программа для задания №2

5. Было выполнено индивидуальное задание. Написанная программа представляет собой программу управления данными о рейсах в словаре. Функция `add_flight()`: добавляет новый рейс в список `destinations`. Пользователь вводит название пункта назначения, номер рейса и тип самолета. Функция `display_flights()`: отображает рейсы для заданного типа самолета. Если не найдено рейсов с указанным типом, выдает соответствующее

сообщение. Функция `print_flight()`: выводит на экран информацию о рейсах в удобном табличном формате.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 # Использовать словарь, содержащий следующие ключи:
5 # название пункта назначения рейса; номер рейса; тип самолета.
6 # Написать программу, выполняющую следующие действия: ввод с клавиатуры
7 # данных в список, состоящий из словарей заданной структуры;
8 # записи должны быть размещены в алфавитном порядке по названию
9 # пункта назначения; вывод на экран пунктов назначения и номеров рейсов,
10 # обслуживаемых самолетом, тип которого введен с клавиатуры;
11 # если таких рейсов нет, вывести на дисплей соответствующее сообщение.
12
13 if __name__ == '__main__':
14     flights = []
15
16     print(">>> select the desired command: add, list, find or exit ")
17
18     while True:
19         command = input(">>> ").lower()
20
21         if command == 'exit':
22             break
23
24         elif command == 'add':
25             destination = input("Введите название пункта назначения: ")
26             flight_number = input("Введите номер рейса: ")
27             plane_type = input("Введите тип самолета: ")
28
29             new_flight = {
30                 'destination': destination,
31                 'flight number': flight_number,
32                 'type of plane': plane_type
33             }
34
35             flights.append(new_flight)
36             flights.sort(key=lambda x: x['destination'])
37
38         elif command == 'list':
39             line = '+--+--+--+--+--+'.format(
40                 '-' * 4,
41                 '-' * 30,
42                 '-' * 20,
43                 '-' * 20
44             )
45             print(line)
46
47             print(
48                 '{:^4} | {:^30} | {:^20} | {:^20}'.format(
49                     "No",
50                     "Пункт назначения",
51                     "Номер рейса",
52                     "Тип самолета"
53                 )
54             )
55
56             print(line)
57
58             for idx, flight in enumerate(flights, 1):
59                 print(
60                     '{:^4} | {:^30} | {:^20} | {:^20}'.format(
61                         idx,
62                         flight.get('destination', ''),
63                         flight.get('flight number', ''),
64                         flight.get('type of plane', 0)
65                     )
66                 )
67             print(line)
68
69         elif command == 'find':
70             find_type = input("Enter the type of plane to search... ")
71             found = []
72
73             for flight in flights:
74                 if flight['type of plane'] == find_type:
75                     found.append(flight)
76
77             if not found:
78                 print("Рейсов на самолете типа '{find_type}' не найдено.")
79             else:
80                 line = '+--+--+--+--+--+'.format(
81                     '-' * 4,
82                     '-' * 30,
83                     '-' * 20,
84                     '-' * 20
85                 )
86                 print(line)
87
88                 print(
89                     '{:^4} | {:^30} | {:^20} | {:^20}'.format(
90                         "No",
91                         "Пункт назначения",
92                         "Номер рейса",
93                         "Тип самолета"
94                     )
95                 )
96
97                 print(line)
98
99                 for idx, found_flight in enumerate(found, 1):
100                     print(
101                         '{:^4} | {:^30} | {:^20} | {:^20}'.format(
102                             idx,
103                             found_flight.get('destination', ''),
104                             found_flight.get('flight number', ''),
105                             found_flight.get('type of plane', 0)
106                         )
107                     )
108                 print(line)
```

Рисунок 5. Программа индивидуального задания

```
>>> select the desired command: add, list, find or exit
>>> add
Введите название пункта назначения: tallinn
Введите номер рейса: 53621
Введите тип самолета: jet
>>> add
Введите название пункта назначения: moscow
Введите номер рейса: 09132
Введите тип самолета: boeing
>>> add
Введите название пункта назначения: rome
Введите номер рейса: 76182
Введите тип самолета: tu134
>>> add
Введите название пункта назначения: tokiio
Введите номер рейса: 62711
Введите тип самолета: jet
>>> list
+--+--+--+--+--+
| No | Пункт назначения | Номер рейса | Тип самолета |
+--+--+--+--+--+
| 1 | moscow | 09132 | boeing |
| 2 | rome | 76182 | tu134 |
| 3 | tallinn | 53621 | jet |
| 4 | tokiio | 62711 | jet |
+--+--+--+--+--+

>>> find
Enter the type of plane to search... jet
+--+--+--+--+--+
| No | Пункт назначения | Номер рейса | Тип самолета |
+--+--+--+--+--+
| 1 | tallinn | 53621 | jet |
| 2 | tokiio | 62711 | jet |
+--+--+--+--+--+

>>> exit
(base) mikheeva@MacBook-Pro-Elena Python_2.6 %
```

Рисунок 6. Результат выполнения индивидуального задания

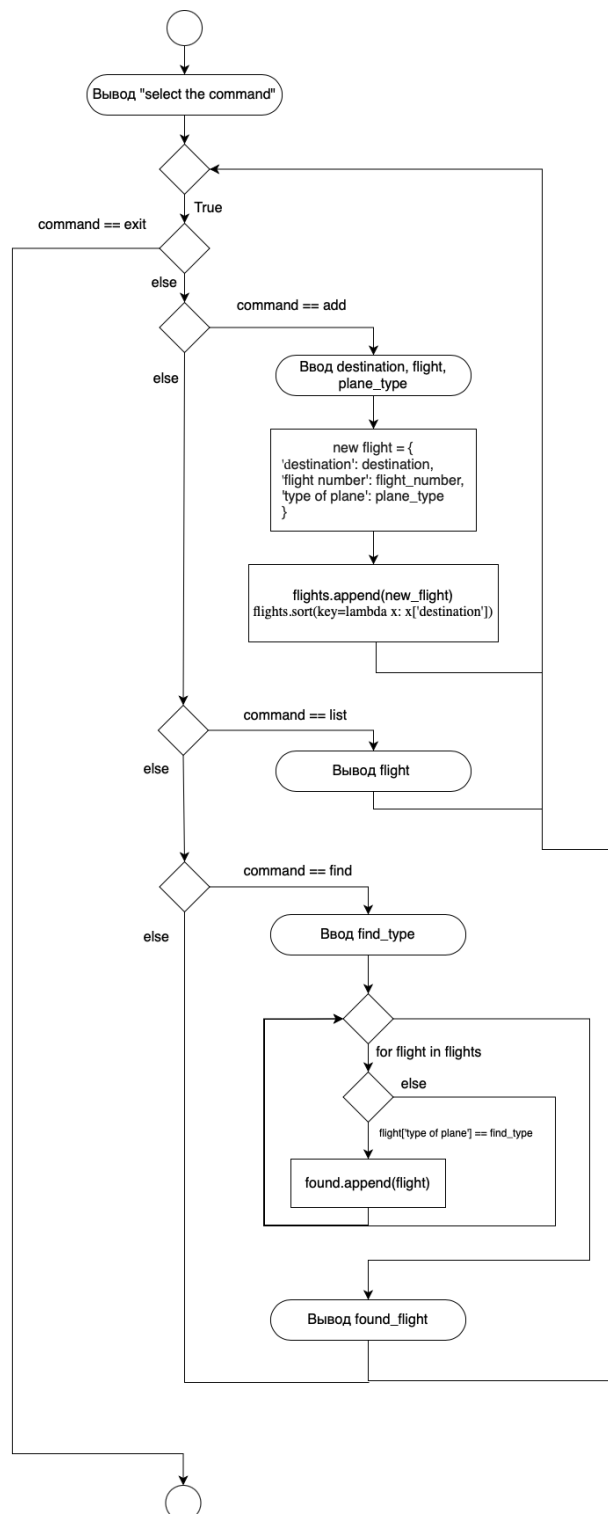


Рисунок 7. UML-диаграмма для индивидуального задания

Ответы на контрольные вопросы.

1. Что такое словари в языке Python?

В языке Python словари – это изменяемые коллекции, которые хранят данные в формате ключ-значение. Каждый элемент словаря состоит из уникального ключа и соответствующего ему значения. Словари в Python

являются неупорядоченными, что означает, что элементы в словаре не имеют определенного порядка. Словари в Python представлены фигурными скобками {} и могут содержать любые типы данных в качестве значений.

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

Да, функция len() может быть использована для определения количества элементов в словаре. Она возвращает количество пар ключ-значение в словаре.

3. Какие методы обхода словарей Вам известны?

Цикл for-in: можно использовать цикл for-in для итерации по ключам словаря и получения соответствующих значений. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
for key in my_dict:
    value = my_dict[key]
    print(key, value)
```

Методы keys(), values() и items(): Метод keys() возвращает итерируемый объект, содержащий все ключи словаря. Метод values() возвращает итерируемый объект, содержащий все значения словаря. Метод items() возвращает итерируемый объект, содержащий кортежи (ключ, значение) для каждой пары ключ-значение в словаре. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
for key in my_dict.keys():
    print(key)
for value in my_dict.values():
    print(value)
for key, value in my_dict.items():
    print(key, value)
```

4. Какими способами можно получить значения из словаря по ключу? В Python можно получить значения из словаря по ключу с помощью следующих способов:

Использование оператора доступа к элементам словаря ([]): Можно использовать оператор доступа к элементам словаря, указав ключ в квадратных скобках. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
print(my_dict['apple'])
```

Метод get(): Метод get() позволяет получить значение по ключу. Если ключ не существует, метод возвращает значение по умолчанию (None, если не указано другое значение). Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
print(my_dict.get('apple'))
```

5. Какими способами можно установить значение в словаре по ключу? В Python можно установить значение в словаре по ключу с помощью следующих способов:

Использование оператора доступа к элементам словаря ([]): Можно использовать оператор доступа к элементам словаря, указав ключ в квадратных скобках и присвоив новое значение. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
my_dict['apple'] = 4
print(my_dict)
```

Метод update(): Метод update() позволяет обновить словарь, добавив новую пару ключ-значение или обновив значение существующего ключа. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
my_dict.update({'apple': 4, 'mango': 6})
print(my_dict)
```

6. Что такое словарь включений?

Словарь включений (dictionary comprehension) - это способ создания словаря на основе итерации и условий. Он позволяет создавать словари более компактно и эффективно. Синтаксис словаря включений выглядит

следующим образом: {ключ\_выражение: значение\_выражение for элемент in итерируемый\_объект if условие}

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

Функция `zip()` в Python используется для объединения элементов из нескольких итерируемых объектов в один итерируемый объект, состоящий из кортежей. Количество элементов в результирующем итерируемом объекте равно количеству элементов в самом коротком итерируемом объекте.

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

Модуль `datetime` в Python предоставляет функционал для работы с датой и временем. Некоторые возможности модуля `datetime` включают: Создание объектов даты и времени: Модуль `datetime` позволяет создавать объекты для представления даты, времени или комбинации даты и времени.