

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:

Мизин Глеб Егорович

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.

Проработка примера 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 соответствии с командой.
17         if command == 'exit':
18             break
19
20         elif command == 'add':
21             # Запросить данные о работнике.
22             name = input("Фамилия и инициалы? ")
23             post = input("Должность? ")
24             year = int(input("Год поступления? "))
25
26             # Создать словарь.
27             worker = {
28                 'name': name,
29                 'post': post,
30                 'year': year,
31             }
32
33             # Добавить словарь в список.
34             workers.append(worker)
35             # Отсортировать список в случае необходимости.
```

Рисунок 1.1 – Код примера

```

36         if len(workers) > 1:
37             workers.sort(key=lambda item: item.get('name', ''))
38
39     elif command == 'list':
40         # Заголовок таблицы.
41         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
42             '-' * 4,
43             '-' * 30,
44             '-' * 20,
45             '-' * 8
46         )
47         print(line)
48         print(
49             '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
50                 "№",
51                 "Ф.И.О.",
52                 "Должность",
53                 "Год"
54             )
55         )
56         print(line)
57
58         # Вывести данные о всех сотрудниках.
59         for idx, worker in enumerate(workers, 1):
60             print(
61                 '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
62                     idx,
63                     worker.get('name', ''),
64                     worker.get('post', ''),
65                     worker.get('year', 0)
66                 )
67             )
68         print(line)
69

```

Рисунок 1.2 – Код примера

```

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
89     # Если счетчик равен 0, то работники не найдены.
90     if count == 0:
91         print("Работники с заданным стажем не найдены.")
92
93 elif command == 'help':
94     # Вывести справку о работе с программой.
95     print("Список команд:\n")
96     print("add - добавить работника;")
97     print("list - вывести список работников;")
98     print("select <стаж> - запросить работников со стажем;")
99     print("help - отобразить справку;")
100    print("exit - завершить работу с программой.")

```

Рисунок 1.3 – Код примера

```

102 else:
103     print(f"Неизвестная команда {command}", file=sys.stderr)
104

```

Рисунок 1.4 – Код примера

```

>>> 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? Евгенийев И.В.
Должность? Директор
Год поступления? 1944
>>> add
Фамилия и инициалы? Мизин Г.Е.
Должность? Попуск
Год поступления? 2021
>>> list
+-----+-----+-----+-----+
|  №   |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|    1  | Евгенийев И.В.          |    Директор         |    1944    |
|    2  | Мизин Г.Е.              |    Попуск           |    2021    |
+-----+-----+-----+-----+
>>> select 1
      1: Евгенийев И.В.
      2: Мизин Г.Е.
>>> seect 1
>>> Неизвестная команда seect 1
exit

Process finished with exit code 0

```

Рисунок 1.4 – Результат работы примера

Задание №1: решите задачу: 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.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5
6     # Source Dictionary
7     school = {
8         '1а': 22,
9         '1б': 23,
10        '1в': 25,
11        '3а': 30,
12        '3е': 24,
13        '7в': 27,
14        '10в': 17,
15        '9а': 31,
16        '3д': 24,
17    }
18
19    # Infinite Command Request Loop
20    while True:
21        command = input("Enter command:")
22        if command == "change":
23            while True:
24                cc = input("Enter class you want change the num of students: ")
25                nn = input("Enter the number of students students: ")
26                if cc in school:
27                    school[cc] = nn
28                break
```

Рисунок 2.1 – Код программы задания №1

```

29         else:
30             print("This class does not exist")
31         print(school)
32
33     elif command == "delete":
34         while True:
35             cc = input("Enter the disbanded class: ")
36             if cc in school:
37                 school.pop(cc)
38                 break
39             else:
40                 print("This class does not exist")
41         print(school)
42
43     elif command == "add":
44         while True:
45             cc = input("Enter new class: ")
46             nn = input("Enter number of students: ")
47             if cc in school:
48                 print("This class already exist")
49             else:
50                 school[cc] = nn
51             break
52         print(school)
53
54     elif command == "exit":
55         break
56

```

Рисунок 2.2 – Код программы задания №1

```

57         else:
58             print("Wrong command!")
59

```

Рисунок 2.3 – Код программы задания №1

```

F:\GitLab\Lab-2.6\venv\Scripts\python.exe F:\GitLab\Lab-2.6\PyCharm\1_task.py
Enter command:change
Enter class you want change the num of students: 1a
Enter the number of students students: 33
{'1a': '33', '16': 23, '1b': 25, '3a': 30, '3e': 24, '7b': 27, '10b': 17, '9a': 31, '3д': 24}
Enter command:delete
Enter the disbanded class: 10a
{'1a': '33', '16': 23, '1b': 25, '3a': 30, '3e': 24, '7b': 27, '9a': 31, '3д': 24}
Enter command:add
Enter new class: 10a
Enter number of students: 15
{'1a': '33', '16': 23, '1b': 25, '3a': 30, '3e': 24, '7b': 27, '9a': 31, '3д': 24, '10г': '15'}
Enter command:exit

Process finished with exit code 0

```

Рисунок 2.4 – Результат работы кода задания №1

Задание №2: решите задачу: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод `items()`, с помощью полученного объекта `dict_items` создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5
6      # Dictionary with keys "numbers" and values "words"
7      first_dict = {
8          1: 'one',
9          2: 'two',
10         3: 'three',
11         4: 'four',
12         5: 'five'
13     }
14
15     # Using the .items function
16     dict_items = first_dict.items()
17
18     # Creating a new dictionary in which keys and values are changed
19     new_dict = {v: k for k, v in dict_items}
20
21     # Output
22     print(new_dict)

```

1_task (1) × 2_task (1) ×

```

F:\GitLab\Lab-2.6\venv\Scripts\python.exe F:\GitLab\Lab-2.6\PyCharm\2_task.py
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}

Process finished with exit code 0

```

Рисунок 3 – Код и результат работы программы задания №2

Индивидуальное задание: использовать словарь, содержащий следующие ключи: расчетный счет плательщика; расчетный счет получателя; перечисляемая сумма в руб. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены в алфавитном порядке по расчетным счетам плательщиков; вывод на экран информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры; если такого расчетного счета нет, выдать на дисплей соответствующее сообщение.

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6
7 ▶ if __name__ == '__main__':
8
9     # List for dictionaries with bank accounts
10    requisites = []
11
12    # Endless program call
13    while True:
14
15        # Entering the required command
16        command = input("Enter Command: ").lower()
17
18        # Command to end the program
19        if command == "exit":
20            break
21
22        # Command to add bank accounts
23        elif command == "add":
24
25            while True:
26                s_b_a = input("Enter the sender's bank account: ")
27
28                # Checking the spelling of a bank account
29                if len(s_b_a) != 20 or s_b_a.isdigit() is False:
30                    print("Incorrect bank account!")
31                else:
32                    break
33
34            while True:
35                b_a = input("Enter the beneficiary's account: ")
36
```

Рисунок 4.1 – Код программы индивидуального задания

```

37         # Checking the spelling of a bank account
38         if len(b_a) != 20 or b_a.isdigit() is False:
39             print("Incorrect bank account!")
40         else:
41             break
42
43     t_a = input("Enter transfer amount in ₺: ")
44
45     # Creating dictionary with bank account
46     requisite = {
47         "s_b_a": s_b_a,
48         "b_a": b_a,
49         "t_a": t_a,
50     }
51
52     # Adding to the list
53     requisites.append(requisite)
54
55     # Sorting
56     if len(requisites) > 1:
57         requisites.sort(key=lambda item: item.get("s_b_a", ""))
58
59     # Command for displaying entered values
60     elif command == "list":
61
62         # Create a table header
63         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
64             '-' * 2,
65             '-' * 25,
66             '-' * 25,
67             '-' * 10
68         )
69         print(line)
70         print(
71             '| {:^2} | {:^25} | {:^25} | {:^10} |'.format(
72                 "№",
73                 "Sender bank account",
74                 "beneficiary account",
75                 "Amount",

```

Рисунок 4.2 – Код программы индивидуального задания

```

76         )
77     )
78     print(line)
79
80     # Filling the table with entered values
81     for ind, requisite in enumerate(requisites, 1):
82         print(
83             '| {:^2} | {:^25} | {:^25} | {:^10} |'.format(
84                 ind,
85                 requisite.get('s_b_a'),
86                 requisite.get('b_a'),
87                 requisite.get('t_a'),
88             )
89         )
90     print(line)
91
92     # The command to receive the withdrawn amount from bank account
93     elif command.startswith("select "):
94
95         # Splitting input into 2 words
96         parts = command.split(" ", maxsplit=1)
97
98         # Assign value to second word (number of bank account)
99         bank_acc = int(parts[1])
100
101         full_summa = 0
102         for requisite in requisites:
103
104             # Comparison of each dictionary from the list by key "s_b_a" with the entered one
105             if int(requisite.get("s_b_a")) == bank_acc:
106
107                 # Calculation of the withdrawn amount from bank account
108                 full_summa += float(requisite.get("t_a"))
109
110         # Checking for the existence of such an account
111         if full_summa == 0:
112             print("This bank account does not exist")
113         else:
114             print("The sum of all transfers of the entered", end=" ")

```

Рисунок 4.3 – Код программы индивидуального задания

```

115         print(f"bank account is equal to {full_summa}")
116
117     # Help Command
118     elif command == 'help':
119         print("Command List:\n")
120         print("add - Add bank account;")
121         print("list - Display a list of bank accounts;")
122         print("select <bank account> -", end=" ")
123         print("The withdrawn amount from account;")
124         print("help - Display Help;")
125         print("exit - End the program.")
126         print("\n")
127
128     # Incorrect command message
129     else:
130         print(f"Invalid command {command}", file=sys.stderr)
131

```

Рисунок 4.4 – Код программы индивидуального задания

```

Enter Command: help
Command List:

add - Add bank account;
list - Display a list of bank accounts;
select <bank account> - The withdrawn amount from account;
help - Display Help;
exit - End the program.

Enter Command: add
Enter the sender's bank account: 12345678900987654321
Enter the beneficiary's account: 12345678900987653222
Enter transfer amount in ₺: 7897456
Enter Command: add
Enter the sender's bank account: 12345678900987654321
Enter the beneficiary's account: 31231231231231231231
Enter transfer amount in ₺: 46579746
Enter Command: add
Enter the sender's bank account: 87894420984273480972
Enter the beneficiary's account: 12345678900987654321
Enter transfer amount in ₺: 123456578

```

Рисунок 4.5 – Результат работы программы индивидуального задания

```

Enter Command: list
+-----+-----+-----+-----+
| № | Sender bank account | beneficiary account | Amount |
+-----+-----+-----+-----+
| 1 | 12345678900987654321 | 12345678900987653222 | 7897456 |
+-----+-----+-----+-----+
| 2 | 12345678900987654321 | 31231231231231231231 | 46579746 |
+-----+-----+-----+-----+
| 3 | 87894420984273480972 | 12345678900987654321 | 123456578 |
+-----+-----+-----+-----+
Enter Command: select 12345678900987654321
The sum of all transfers of the entered bank account is equal to 54477202.0
Enter Command: seect 12345678900987654321
Invalid command
Enter Command: exit

```

Рисунок 4.6 – Результат работы программы индивидуального задания

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.

Основной пример:

```
>>> {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.