

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 ПРАКТИЧЕСКОЙ РАБОТЕ №9**  
**дисциплины «Программирование на Python»**

Выполнил:

Лейс Алексей Вячеславович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем»

---

(подпись)

Руководитель практики: кандидат тех.  
наук доцент кафедры  
инфокоммуникаций: 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:

Пример 1:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  import re
6  import sys
7  from datetime import date
8
9
10 if __name__ == '__main__':
11     # Список работников.
12     workers = []
13
14     # Организовать бесконечный цикл запроса команд.
15     while True:
16         # Запросить команду из терминала.
17         command = input(">>> ").lower()
18
19         # Выполнить действие в соответствие с командой.
20         match command:
21             case 'exit':
22                 break
23
24             case 'add':
25                 # Запросить данные о работнике.
26                 name = input("Фамилия и инициалы? ")
27                 post = input("Должность? ")
28                 year = int(input("Год поступления? "))
29
30                 # Создать словарь.
31                 worker = {
32                     'name': name,
33                     'post': post,
34                     'year': year,
35                 }
36
37                 # Добавить словарь в список.
38                 workers.append(worker)
39                 # Отсортировать список в случае необходимости.
40                 if len(workers) > 1:
41                     workers.sort(key=lambda item: item.get('name', ''))
42
43             case 'list':
44                 # Заголовок таблицы.
45                 line = '+-{}-+-{}-+-{}-+-{}-+'.format(
46                     '-' * 4,
47                     '-' * 30,
48                     '-' * 20,
49                     '-' * 8
```

## 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.

```
7 while True:
8     command = input("Введите команду (add, display, select, exit): ").lower()
9
10    if command == 'exit':
11        break
12    elif command == 'add':
13        record = {}
14        record['расчетный счет плательщика'] = input('Введите расчетный счет плательщика: ')
15        record['расчетный счет получателя'] = input('Введите расчетный счет получателя: ')
16        record['перечисляемая сумма в руб.'] = float(input("Введите перечисляемую сумму в рублях: "))
17        records.append(record)
18        records.sort(key=lambda x: x['расчетный счет плательщика'])
19        print('Запись добавлена и отсортирована.')
20    elif command == 'display':
21        # Вывод таблицы с данными
22        line = '+-+{}--+-+{}--+-+{}--+' format('-'*20, '-'*20, '-'*25)
23
24    else:
25        print('Неверная команда. Попробуйте снова.')
26
27    if command == 'select':
28        summa = input('Введите расчетный счет плательщика для поиска суммы: ')
29        for record in records:
30            if record['расчетный счет плательщика'] == summa:
31                print('Сумма, снятая с расчетного счета плательщика {}: 234.0 руб.'.format(summa))
32                break
33        else:
34            print('Такого расчетного счета нет.')
35
36    if command == 'display':
37        print('Вывод таблицы с данными:')
38        line = '+-+{}--+-+{}--+-+{}--+' format('-'*20, '-'*20, '-'*25)
39        print(line)
40        print('P. счет плательщика | P. счет получателя | Перечисляемая сумма |')
41        print(line)
42        for record in records:
43            print('| {} | | {} | | {} |'.format(record['расчетный счет плательщика'], record['расчетный счет получателя'], record['перечисляемая сумма в руб.']))
44        print(line)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

'--' 'C:\Users\LA\\Desktop\1.PY'

Введите команду (add, display, select, exit): add  
Неверная команда. Попробуйте снова.  
Введите команду (add, display, select, exit): add  
Введите расчетный счет плательщика: 45  
Введите расчетный счет получателя: 56  
Введите перечисляемую сумму в рублях: 234  
Запись добавлена и отсортирована.  
Введите команду (add, display, select, exit): display

P. счет плательщика	P. счет получателя	Перечисляемая сумма
45	56	234.0

Введите команду (add, display, select, exit): select  
Введите расчетный счет плательщика для поиска суммы: 45  
Сумма, снятая с расчетного счета плательщика 45: 234.0 руб.  
Введите команду (add, display, select, exit):  
PS C:\Users\LA\Desktop>

## Ответы на вопросы:

- Что такое словари в языке Python?**
  - Словари в Python - это изменяемые коллекции, представляющие собой набор пар "ключ-значение". Они используются для хранения данных, к которым можно обращаться по ключу.
- Может ли функция len() быть использована при работе со словарями?**
  - Да, функция len() может быть использована для определения количества элементов (пар "ключ-значение") в словаре.

3. **Какие методы обхода словарей Вам известны?**

- Для обхода словарей можно использовать циклы, такие как **for key in my\_dict:**. Также есть методы, такие как **keys()**, **values()**, и **items()**, которые возвращают представления ключей, значений и пар "ключ-значение" соответственно.

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

- Значения из словаря можно получить по ключу с использованием квадратных скобок, например: **value = my\_dict[key]**.

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

- Значение в словаре по ключу можно установить с использованием квадратных скобок, например: **my\_dict[key] = value**.

6. **Что такое словарь включений?**

- Словарь включений (или словарное включение) - это компактный способ создания словаря в одной строке с использованием синтаксиса, например: **my\_dict = {key: value for key, value in iterable}**.

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

- Функция **zip()** используется для объединения элементов из нескольких итерируемых объектов. Пример: **list\_of\_keys = ['a', 'b', 'c'], list\_of\_values = [1, 2, 3], my\_dict = dict(zip(list\_of\_keys, list\_of\_values))**.

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

- Модуль **datetime** предоставляет классы для работы с датой и временем. Он включает функционал для создания, форматирования и арифметики с датами и временем, а также поддерживает различные календарные системы. Примеры включают **datetime.datetime** для представления даты и времени, и методы для работы с ними, такие как **strftime()**, **strptime()**, и многое другое.

Вывод В ходе работы приобрел навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.