

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

Отчет по лабораторной работе №7

По дисциплине основы кроссплатформенного программирования  
«Работа со словарями в языке Python»

Выполнила:

студентк группы ИТС-б-о-21-1

Абдикодиров Жахонгир Хуснитдин  
угли

---

(подпись)

Проверил: Доцент, к.т.н, доцент  
кафедры

инфокоммуникаций

Воронкин Р. А.

Работа защищена с оценкой:

---

(подпись)

Ставрополь, 2022

## 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.

Написал код для задания 1

```
7LR > ex_1.py > ...
You, 1 секунду назад | 1 author (You)
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      class_in_school = {'1a': 25, '2a': 30, '4a': 24, '5a': 23,
8                          '6a': 21, '7a': 18, '8a': 23, '8a': 15, }
9
10     class_in_school['1a'] = 22
11     class_in_school['3a'] = 21
12     del class_in_school['4a']
13     print(class_in_school)
14
15     f = sum(class_in_school.values())
16     print(f)
```

задание 1

Написал код для задание 2

```
7LR > ex_2.py > ...
You, 4 минуты назад | 1 author (You)
1  #!/usr/bin/env python 3
2  # -*- config: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      dict = {1: "один", 2: "два", 3: "три", 4: "четыре", 5: "пять", }
8      dict_items = {j: i for i, j in dict.items()}
9      print(dict, dict_items)
```

Задание 2

## Написал код для индивидуального задания

```
7LR > 1ind.py > ...
You, 4 минуты назад | 1 author (You)
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Список студентов.
8      students = []
9
10     print("Список команд:\n")
11     print("add - добавить студента;")
12     print("list - вывести список студентов;")
13     print("select <средний балл> - запросить студентов с баллом выше 4.0;")
14     print("exit - завершить работу с программой.")
15
16     # Организовать бесконечный цикл запроса команд.
17     while True:
18         # Запросить команду из терминала.
19         command = input(">>> ").lower()
20
21         # Выполнить действие в соответствие с командой.
22         if command == 'exit':
23             break
```

Активация V

```
25     elif command == 'add':
26         # Запросить данные о студенте.
27         name = input("Фамилия и инициалы? ")
28         group = input("Номер группы? ")
29         grade = list(map(int, input("Успеваемость ").split()))
30         # Создать словарь.
31         student = {
32             'name': name,
33             'group': group,
34             'grade': grade,
35         }
36         # Добавить словарь в список.
37         students.append(student)
38         # Отсортировать список в случае необходимости.
39         if len(students) > 1:
40             students.sort(key=lambda item: item.get('name', ''))
41
42     elif command == 'list':
43         # Заголовок таблицы.
44         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
45             '-' * 4,
46             '-' * 30,
47             '-' * 20,
48             '-' * 15
49         )
```

Активация V  
Чтобы активиро  
раздел "Параме

```
7LR > 1ind.py > ...
50     print(line)
51     print(
52         '| {:^4} | {:^30} | {:^20} | {:^15} |'.format(
53             "№",
54             "Ф.И.О.",
55             "Группа",
56             "Успеваемость"
57         )
58     )
59     print(line)
60
61     # Вывести данные о всех студентах.
62     for idx, student in enumerate(students, 1):
63         print(
64             '| {:>4} | {:<30} | {:<20} | {:>15} |'.format(
65                 idx,
66                 student.get('name', ''),
67                 student.get('group', ''),
68                 student.get('grade', 0)
69             )
70         )
71     print(line)
72     elif command.startswith('select '):
73         # Инициализировать счетчик.
74         count = 0
```

```

75 # Проверить сведения студентов из списка.
76 for student in students:
77     if "4" in student.get('grade', ''):
78         count += 1
79         print(
80             '{:>4}: {}'.format(count, student.get('name', '')),
81             '{:>1} {}'.format('группа №', student.get('group', ''))
82         )
83 # Если счетчик равен 0, то студенты не найдены.
84 if count == 0:
85     print("Студенты с баллом 4.0 и выше не найдены.")
86 else:
87     print(f"Неизвестная команда {command}", file=sys.stderr)
88

```

## Индивидуальное задание

### Вопросы для защиты работы

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

Словарь - это изменяемый (как список) неупорядоченный (в отличие от строк, списков и кортежей) набор элементов "ключ: значение".

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

Может

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

Кроме рассмотренных выше трех методов items(), keys() и values() словари обладают еще восемью. Это методы clear(), copy(), fromkeys(), get(), pop(), popitem(), setdefault(), update().

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

Метод get() позволяет получить элемент по его ключу:

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

Конкретное значение в словаре можно получить, введя название словаря и затем название ключа в квадратных скобках. person ['last name'].

#### 6. Что такое словарь включений?

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

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

Функция zip объединяет элементы различных итерируемых объектов (таких как списки, кортежи или множества) и возвращает итератор.

Вот пример её применения для объединения двух списков: `id = [1, 2, 3, 4]`  
`leaders = ['Elon Mask', 'Tim Cook', 'Bill Gates', 'Yang Zhou']` `record = zip(id, leaders)`.

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

`Datetime` — важный элемент любой программы, написанной на Python. Этот модуль позволяет управлять датами и временем, представляя их в таком виде, в котором пользователи смогут их понимать.

`datetime` включает различные компоненты. Так, он состоит из объектов следующих типов:

- `date` — хранит дату
- `time` — хранит время
- `datetime` — хранит дату и время

### **Вывод:**

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