

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

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 2.6 Работа со
словарями в языке Python»**

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

Выполнил:

Луценко Дмитрий Андреевич
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.

Лабораторная работа 2.6 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.

Ход работы:

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  > import sys
5
6
7  > if __name__ == '__main__':
8  >     school = {}
35
36     while True:
37         command = input(">>> ").lower()
38
39         if command == 'exit':
40             break
41
42         elif command == 'add':
43             class_num = input("Введите число и букву класса: ")
44             pupils = int(input("Кол-во человек? "))
45             school[class_num] = pupils
46
47         elif command == "list":
48             for i, item in enumerate(school):
49                 print(f"Класс {item}, учеников {school[item]}")
50
```

Рисунок 1 – Код программы для задания 1

```
51     elif command == 'delete':
52         while True:
53             del_class = input("Введите удаляемый класс(для выхода exit): ")
54
55             if del_class == 'exit':
56                 break
57
58             elif del_class in school:
59                 school.pop(del_class)
60
61             else:
62                 print("Введённого класса не существует")
63
64         elif command == 'count':
65             count = 0
66             for i, item in enumerate(school):
67                 count += school[item]
68             print(count)
69
```

Рисунок 2 – Код программы для задания 1

```

64 elif command == 'count':
65     count = 0
66     for i, item in enumerate(school):
67         count += school[item]
68     print(count)
69
70 elif command == 'help':
71     print("Список команд:\n")
72     print("add - добавить класс;)")
73     print("list - вывести список школьных классов;)")
74     print("delete - удалить школьный класс;)")
75     print("count - удалить школьный класс;)")
76     print("help - отобразить справку;)")
77     print("exit - завершить работу с программой.")
78
79 else:
80     print(f"Неизвестная команда {command}", file=sys.stderr)

```

Рисунок 3 – Код программы для задания 1

```

>>> add
Введите число и букву класса: 6Д
Кол-во человек? 20
>>> delete
Введите удаляемый класс(для выхода exit): 2В
Введите удаляемый класс(для выхода exit): exit

```

Рисунок 4 – Ввод команд add и delete

```

>>> count
636
>>>

```

Рисунок 5 – Ввод команды count

```
>>> list
Класс 1А, учеников 30
Класс 1Б, учеников 28
Класс 2А, учеников 29
Класс 2Б, учеников 29
Класс 3А, учеников 30
Класс 3Б, учеников 29
Класс 4А, учеников 28
Класс 4Б, учеников 29
Класс 4В, учеников 25
Класс 5А, учеников 27
Класс 5Б, учеников 25
Класс 6А, учеников 29
Класс 6Б, учеников 27
Класс 6В, учеников 23
Класс 7А, учеников 32
Класс 8А, учеников 30
Класс 8Б, учеников 19
Класс 9А, учеников 29
Класс 9Б, учеников 27
Класс 9В, учеников 21
Класс 10А, учеников 22
Класс 10Б, учеников 15
Класс 11А, учеников 17
Класс 11Б, учеников 16
Класс 6Д, учеников 20
>>> |
```

Рисунок 6 – Результат команды list после одного добавленного класса и одного удалённого

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  import sys
5
6
7  ▶  if __name__ == '__main__':
8      dictionary = {1: 'бѣбра', 2: 'СКФУ', 3: 'Питон'}
9
10     swapped = {v: k for k, v in dictionary.items()}
11
12     print(swapped)
```

Рисунок 7 – Код программы для задания 2

Индивидуальное задание

18. Использовать словарь, содержащий следующие ключи: название товара; название магазина, в котором продается товар; стоимость товара 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 названиям магазинов; вывод на экран информации о товарах, продающихся в магазине, название которого введено с клавиатуры; если такого магазина 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       products = []
9
10      while True:
11          command = input(">>> ").lower()
12
13          if command == 'exit':
14              break
15
16          elif command == 'add':
17              prod = input("Введите название товара: ")
18              shop = input("Введите название магазина: ")
19              cost = float(input("Введите стоимость товара: "))
20              product_ = {
21                  'product': prod,
22                  'shop': shop,
23                  'cost': cost
24              }
25
26              products.append(product_)
27              if len(products) > 1:
28                  products.sort(key=lambda item: item.get('shop', ''))
29
```

Рисунок 8 – Код программы для индивидуального задания

```

28         products.sort(key=lambda item: item.get('shop', ''))
29
30     elif command == 'list':
31         line = '{}-{}--{}--{}-{}'.format(
32             '-' * 4,
33             '-' * 30,
34             '-' * 20
35         )
36         print(line)
37         print(
38             '| {:^25} | {:^15} | {:^14} |'.format(
39                 "Товар",
40                 "Магазин",
41                 "Стоимость"
42             )
43         )
44         print(line)
45
46         for product in products:
47             print(
48                 '| {:^25} | {:^15} | {:^14} |'.format(
49                     product.get('product', ''),
50                     product.get('shop'+'_'),
51                     product.get('cost'+'_')
52                 )
53             )
54         print(line)
55
56     elif command == 'select':

```

Рисунок 9 – Код программы для индивидуального задания

```

55
56     elif command == 'select':
57         sel_shop = input("Введите магазин: ")
58
59         for product in products:
60             if product.get('shop', '') == sel_shop:
61                 print(
62                     '| {:^25} | {:^15} | {:^14} |'.format(
63                         product.get('product', ''),
64                         product.get('shop', ''),
65                         product.get('cost', '')
66                     )
67                 )
68
69             else:
70                 print("Магазин не найден!")
71
72     elif command == 'help':
73         print("Список команд:\n")
74         print("add - добавить информацию о товаре;")
75         print("list - вывести список товаров;")
76         print("select - запросить товары из одного магазина;")
77         print("help - отобразить справку;")
78         print("exit - завершить работу с программой.")
79
80     else:
81         print(f"Неизвестная команда {command}", file=sys.stderr)
82

```

Рисунок 10 – Код программы для индивидуального задания

```

Введите название товара: макароны
Введите название магазина: магнит
Введите стоимость товара: 40
>>> add
Введите название товара: телефон
Введите название магазина: ситилинк
Введите стоимость товара: 20000
>>> list
+-----+-----+-----+
|          Товар          | Магазин | Стоимость |
+-----+-----+-----+
|      ваз 2170      |   лада   | 500000.0 |
|      гречка      |   магнит  |    80.0   |
|      макароны      |   магнит  |    40.0   |
|      кола      | пятерочка |    120.0   |
|      телефон      | ситилинк  | 20000.0   |
+-----+-----+-----+
>>>

```

Рисунок 11 – Результат команд add и list программы для индивидуального задания

```

>>> select
Введите магазин: магнит
|          гречка          |    магнит    |    80.0    |
|        макароны        |    магнит    |    40.0    |
>>> select
Введите магазин: днс
Магазин не найден!
>>>

```

Рисунок 12 – Результат команды select программы для индивидуального задания

Вывод: были приобретены навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.

Ответы на 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.

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.