

и 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 31

Выполнил:
Репкин Александр Павлович
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», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р.А., канд. техн. наук,
доцент кафедры инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____
Ставрополь, 2023 г.

Тема: Работа со словарями в языке Python.

Цель: приобрести навыки работы со словарями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Выполнен пример. В нём создавался словарь, хранящий информацию о сотрудниках – ФИО, должность, год поступления на работу. Требовалось обрабатывать несколько команд – вывод всех сотрудников (list), вывод сводки о имеющихся командах (help), добавление сотрудника (add), вывод сотрудников относительно требуемой продолжительности работы (select “срок”), выход из программы (exit).

```
>>> list
+-----+-----+-----+-----+
| № |           Ф.И.О.           |      Должность      |      Год      |
+-----+-----+-----+-----+
>>> add
Фамилия и инициалы? Колоколов П.Р.
Должность? Заместитель
Год поступления? 1975
>>> add
Фамилия и инициалы? Прознавший Ж.Т.
Должность? Следователь
Год поступления? 1991
>>> add
Фамилия и инициалы? Новаторов К.Р.
Должность? Исследователь
Год поступления? 1992
>>> select 32
1: Колоколов П.Р.
2: Прознавший Ж.Т.
>>> select 31
1: Колоколов П.Р.
2: Новаторов К.Р.
3: Прознавший Ж.Т.
>>> list
+-----+-----+-----+-----+
| № |           Ф.И.О.           |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Колоколов П.Р.           |      Заместитель     |      1975     |
|  2 | Новаторов К.Р.           |      Исследователь   |      1992     |
|  3 | Прознавший Ж.Т.         |      Следователь     |      1991     |
+-----+-----+-----+-----+
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
```

Рисунок 1. Полученный результат примера.

2. Выполнено задание №9 - требовалось создать 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.). Затем было необходимо внести 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. В конце выводилось общее количество учащихся в школе.

```
Good day, our current list of classes is:
{'1A': 12, '1B': 21, '2A': 45, '2B': 23, '4A': 32, '5C': 14, '6F': 22}
Please, give us the letter and the number of class, in which amount of students has changed - 2A
How many students are there now? - 12
Current list of classes:
{'1A': 12, '1B': 21, '2A': 12, '2B': 23, '4A': 32, '5C': 14, '6F': 22}
Now give us the number and letter of a new class - 7F
Amount of students in it - 33
Which class doesn't exist anymore? - 4A
Final variant of list of classes:
{'1A': 12, '1B': 21, '2A': 12, '2B': 23, '5C': 14, '6F': 22, '7F': 33}
Overall amount of students - 137
```

Рисунок 2. Полученный результат задания №9.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  if __name__ == "__main__":
6      school = {"1A": 12, "1B": 21, "2A": 45,
7               "2B": 23, "4A": 32, "5C": 14, "6F": 22}
8      print("Good day, our current list of classes is:\n", school)
9      # Задание а - изменить число студентов в требуемом классе.
10     changed_class_key = input(
11         "Please, give us the letter and the number of class, in which amount of students has changed - ")
12     changed_class_value = int(input("How many students are there now? - "))
13     if school[changed_class_key] == changed_class_value:
14         print("Excuse us, but amount of students hasn't changed.")
15     else:
16         school[changed_class_key] = changed_class_value
17         print("Current list of classes:\n", school)
18     # Задание б - добавить новый класс и информацию о нём.
19     new_class_key = input(
20         "Now give us the number and letter of a new class - ")
21     new_class_value = int(input("Amount of students in it - "))
22     school[new_class_key] = new_class_value
23     # Задание с - удалить класс.
24     goodbye = input("Which class doesn't exist anymore? - ")
25     del school[goodbye]
26     print("Final variant of list of classes:\n", school,
27           "\nOverall amount of students - ", sum(school.values()))
28
```

Рисунок 3. Полученный код задания №9.

3. Выполнено задание №11 - требовалось создать словарь, где ключами являются числа, а значениями – строки. Применив к нему метод items(), и с помощью полученного объекта dict_items создан новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.

```
Good day, first variant of dictionary:
{45: '-', 46: '.', 47: '/', 48: '0', 49: '1', 50: '2', 51: '3', 52: '4', 53: '5', 54: '6'}
Reversed variant of dictionary:
{'-': 45, '.': 46, '/': 47, '0': 48, '1': 49, '2': 50, '3': 51, '4': 52, '5': 53, '6': 54}
```

Рисунок 4. Полученный результат задания №11.

```
1  #!usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  if __name__ == "__main__":
6      # Первый словарь, где ключи - цифры, а значения - строки.
7      key_numbers = {i: chr(i) for i in range(45, 55)}
8      print("Good day, first variant of dictionary:\n", key_numbers)
9      # Второй словарь, где ключи - строки, а значения - символы.
10     key_symbols = {i: j for j, i in key_numbers.items()}
11     print("Reversed variant of dictionary:\n", key_symbols)
12
```

Рисунок 5. Полученный код задания №11.

4. Выполнено индивидуальное задание, в нём создавался словарь, хранящий информацию о людях – фамилия, имя, телефонный номер, дата рождения. Требовалось обрабатывать несколько команд – вывод всех людей (list), вывод сводки о имеющихся командах (help), добавление человека (add), вывод людей относительно требуемого месяца рождения (select “месяц”), выход из программы (exit).

```
Good day!, please, enter your command: ('help' will list all of them)
>>> list
+-----+-----+-----+-----+-----+
| № | Name | Surname | Telephone | Birthday |
+-----+-----+-----+-----+-----+
>>> add
Name - Afon
Surname - Back
Telephone number - 88005553535
Date of birth (Day.Month.Year) - 09.12.2015
>>> add
Name - Abark
Surname - Abon
Telephone number - 85401234567
Date of birth (Day.Month.Year) - 12.03.2012
>>> add
Name - Ion
Surname - Aagar
Telephone number - 86503230987
Date of birth (Day.Month.Year) - 2.3.1927
>>> list
+-----+-----+-----+-----+-----+
| № | Name | Surname | Telephone | Birthday |
+-----+-----+-----+-----+-----+
| 1 | Ion | Aagar | 86503230987 | 2.3.1927 |
| 2 | Abark | Abon | 85401234567 | 12.03.2012 |
| 3 | Afon | Back | 88005553535 | 09.12.2015 |
+-----+-----+-----+-----+-----+
>>> select 9
There is no such a human.
>>> select 12
1: Afon Back
>>> select 3
1: Ion Aagar
2: Abark Abon
>>> exit
```

Рисунок 6. Полученный результат индивидуального задания.

Ответ: Словарь 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. В списке доступ к элементам осуществляется по индексу, который представляет собой целое неотрицательное число. В словаре аналогом индекса является ключ, при этом ответственность за его формирование ложится на программиста. В языке программирования Python словари (тип dict) представляют собой еще одну разновидность структур данных наряду со списками и кортежами. Словарь - это изменяемый (как список) неупорядоченный (в отличие от строк, списков и кортежей) набор элементов "ключ: значение". ("Неупорядоченный" – значит, что последовательность расположения пар не важна. Язык программирования ее не учитывает, в следствие чего обращение к элементам по индексам невозможно).

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

Ответ: функция len() выводит количество ключей в словаре.

3) Какие методы обхода словарей Вам известны?

Ответ: для обхода словаря можно воспользоваться циклом for с оператором in, а также с методами keys() (Все ключи), items() (Превращает пары ключ-значение в отдельные списки в кортеже) или values() (Все значения).

4) Какими способами можно получить значения из словаря по ключу?

Ответ: объект словаря обладает функцией get(), которой можно пользоваться для доступа к элементам словаря. Ее нужно добавлять к словарю через точку и затем передавать название ключа как аргумент функции.

5) Какими способами можно установить значение в словаре по ключу?

Ответ: чтобы в словарь добавить значение ключу, можно: `map = {1: 'one', 2: 'two', 3: 'three'}; nums2[4] = 'four'; nums.setdefault(4, 'four')`.

6) Что такое словарь включений?

Ответ: словарное включение работает по принципу: {ключ: значение for элемент in итерируемый_объект}. Словарное включение аналогично списковым включениям, то есть сокращает длину кода, позволяя заполнить словарь всего одной строкой кода.

7) Возможности функции `zip()`? Приведите примеры ее использования.

Ответ: функция `zip()` в Python позволяет объединять элементы из нескольких итерируемых объектов (списков, строк) в кортежи. Каждый кортеж содержит элементы на соответствующих позициях из каждого переданного итерируемого объекта. Если объекты имеют разную длину, то `zip()` остановится, когда закончится итерация на самом коротком объекте. Например: `first = [2, 9, 18, 28]; second = ["Smth1", "Hey!", "Good day", "2htmS"];`
`zip(first, second) ==> [(2, "Smth1"), (9, "Hey!"), (18, "Good day"), (28, "2htmS")]`.

8) Возможности модуля `datetime`? Каким функционалом по работе с датой и временем обладает этот модуль?

Ответ: Этот модуль позволяет управлять датами и временем, представляя их в таком виде, в котором пользователи смогут их понимать. `datetime` включает различные компоненты (Можно переставлять местами вывод месяца, года и дня, изменять часовой пояс, указывать 12-ти или 24-часовой формат и пр.). Так, он состоит из объектов следующих типов: `date` — хранит дату `time` — хранит время `datetime` — хранит дату и время.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки работы со словарями при написании программ с помощью языка программирования Python версии 3.x.