

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 №2.6
дисциплины «Программирование на Python»

Выполнил:
Степанов Леонид Викторович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение
средств вычислительной техники
и автоматизирование систем»,
очная форма обучения

(подпись)

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

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа со словарями в языке Python

Цель: приобретение навыков по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Создал файл (primer.py), в котором проработал пример 1, в нем эмулируется консоль, в которой можно добавлять работников, просматривать их, выбирать сотрудника с определённым стажем, выводить список команд.

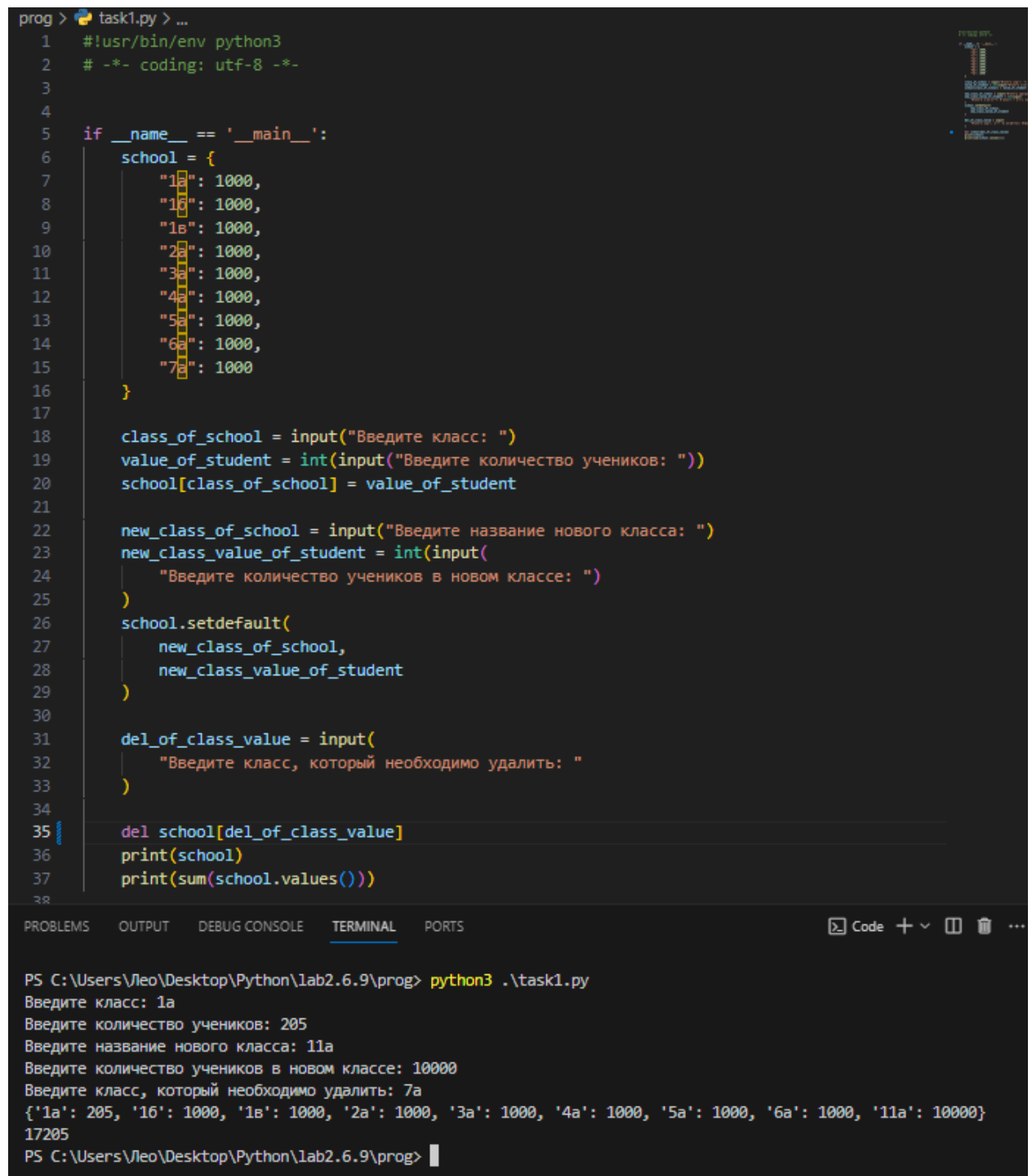
```
from distutils import command
>>> add
Фамилия и инициалы? Степанов Л.В.
Должность? Студент
Год поступления? 2021
>>> add
Фамилия и инициалы? Иванов И.И.
Должность? Администратор
Год поступления? 2010
>>> list
+-----+-----+-----+-----+
| № |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Иванов И.И.              | Администратор       |      2010     |
|  2 | Степанов Л.В.           | Студент             |      2021     |
+-----+-----+-----+-----+
>>> select 4
1: Иванов И.И.
>>> select 2
1: Иванов И.И.
2: Степанов Л.В.
>>> select 11
1: Иванов И.И.
>>> select 40
Работники с заданным стажем не найдены.
>>> help
Список команд:

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

Рисунок 1 – Результат выполнения программы primer.py

2. Создал файл (task1.py), в котором решил 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.



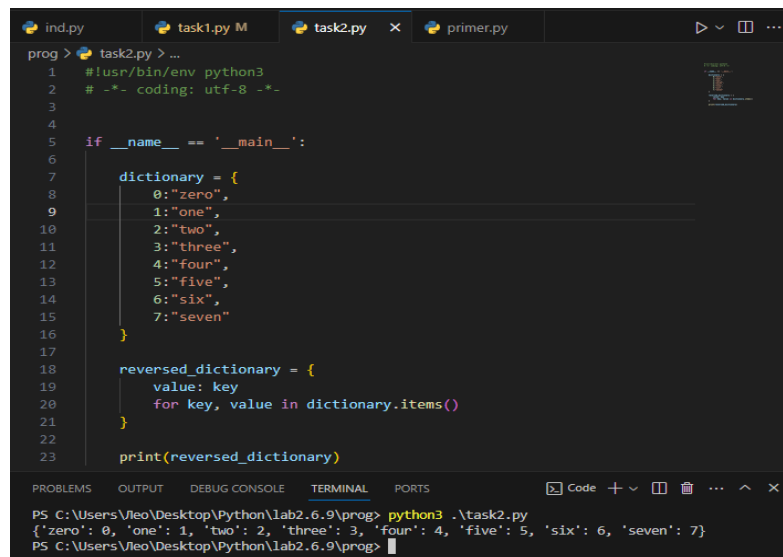
```
prog > task1.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  if __name__ == '__main__':
6      school = {
7          "1a": 1000,
8          "1б": 1000,
9          "1в": 1000,
10         "2a": 1000,
11         "3a": 1000,
12         "4a": 1000,
13         "5a": 1000,
14         "6a": 1000,
15         "7a": 1000
16     }
17
18     class_of_school = input("Введите класс: ")
19     value_of_student = int(input("Введите количество учеников: "))
20     school[class_of_school] = value_of_student
21
22     new_class_of_school = input("Введите название нового класса: ")
23     new_class_value_of_student = int(input(
24         "Введите количество учеников в новом классе: ")
25     )
26     school.setdefault(
27         new_class_of_school,
28         new_class_value_of_student
29     )
30
31     del_of_class_value = input(
32         "Введите класс, который необходимо удалить: "
33     )
34
35     del school[del_of_class_value]
36     print(school)
37     print(sum(school.values()))
38
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Leo\Desktop\Python\lab2.6.9\prog> python3 .\task1.py
Введите класс: 1a
Введите количество учеников: 205
Введите название нового класса: 11a
Введите количество учеников в новом классе: 10000
Введите класс, который необходимо удалить: 7a
{'1a': 205, '1б': 1000, '1в': 1000, '2a': 1000, '3a': 1000, '4a': 1000, '5a': 1000, '6a': 1000, '11a': 10000}
17205
PS C:\Users\Leo\Desktop\Python\lab2.6.9\prog> |

Рисунок 2 – Результат выполнения программы task1.py

3. Создал файл (task2.py), в котором решил 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.



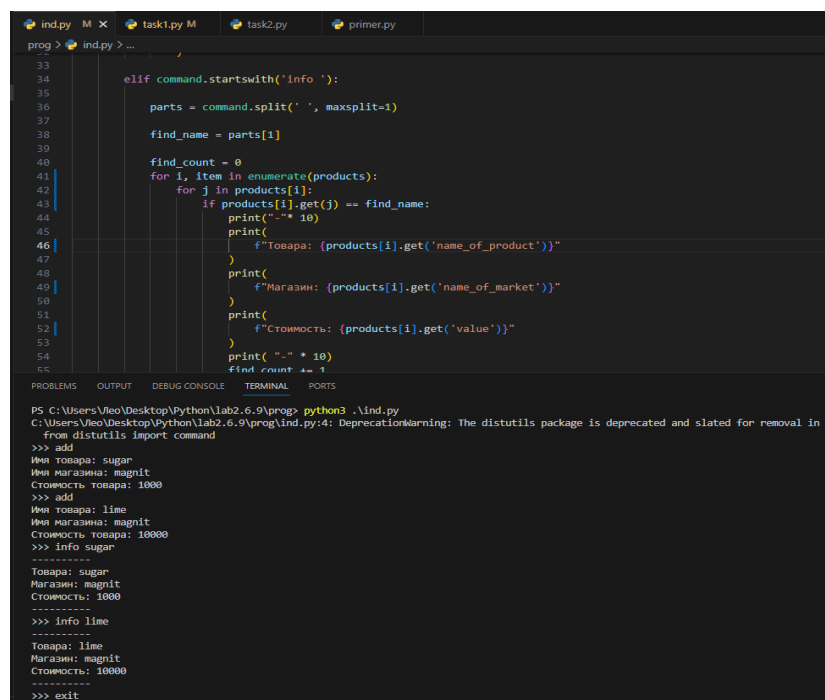
```
1 #!usr/bin/env python3
2 #-*- coding: utf-8 -*-
3
4
5 if __name__ == '__main__':
6
7     dictionary = {
8         0:"zero",
9         1:"one",
10        2:"two",
11        3:"three",
12        4:"four",
13        5:"five",
14        6:"six",
15        7:"seven"
16    }
17
18    reversed_dictionary = {
19        value: key
20        for key, value in dictionary.items()
21    }
22
23    print(reversed_dictionary)
```

PS C:\Users\Ileo\Desktop\Python\lab2.6.9\prog> python3 .\task2.py
{'zero': 0, 'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5, 'six': 6, 'seven': 7}

Рисунок 3 – Результат выполнения программы task1.py

Индивидуальное задание (17 Вариант):

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 товаров; вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать на дисплей соответствующее сообщение.



```
33
34 elif command.startswith('info '):
35
36     parts = command.split(' ', maxsplit=1)
37
38     find_name = parts[1]
39
40     find_count = 0
41     for i, item in enumerate(products):
42         for j in products[i]:
43             if products[i].get(j) == find_name:
44                 print("-" * 10)
45                 print(
46                     f"Товара: {products[i].get('name_of_product')}"
47                 )
48                 print(
49                     f"Магазин: {products[i].get('name_of_market')}"
50                 )
51                 print(
52                     f"Стоимость: {products[i].get('value')}"
53                 )
54                 print("-" * 10)
55                 find_count += 1
```

PS C:\Users\Ileo\Desktop\Python\lab2.6.9\prog> python3 .\ind.py
C:\Users\Ileo\Desktop\Python\lab2.6.9\prog\ind.py:4: DeprecationWarning: The distutils package is deprecated and slated for removal in P
from distutils import command
>>> add
Имя товара: sugar
Имя магазина: magnit
Стоимость товара: 1000
>>> add
Имя товара: lime
Имя магазина: magnit
Стоимость товара: 10000
>>> info sugar

Товара: sugar
Магазин: magnit
Стоимость: 1000

>>> info lime

Товара: lime
Магазин: magnit
Стоимость: 10000

>>> exit

Рисунок 4 – Результат выполнения программы ind.py

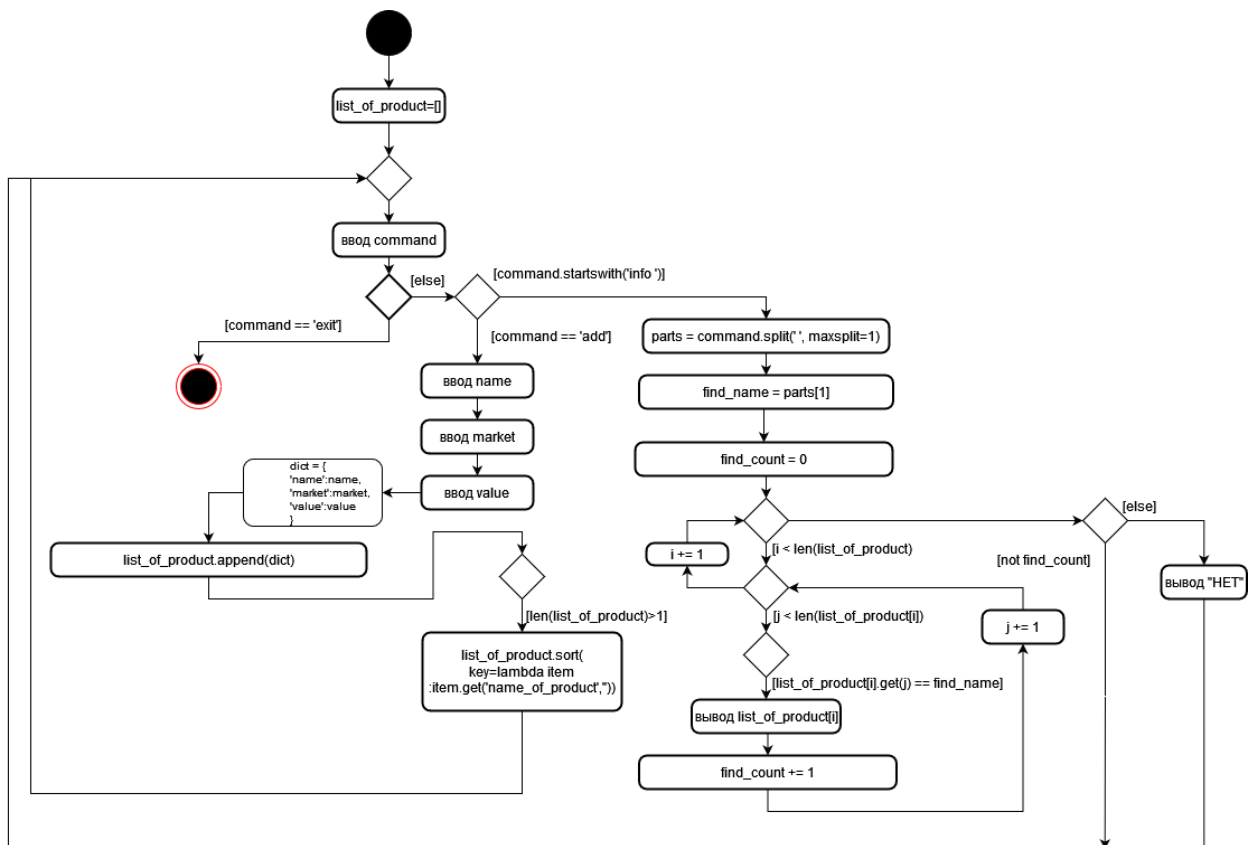


Рисунок 5 – UML-диаграмма программы ind.py

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

Контрольные вопросы

1. Что такое словари в языке Python?

В языке Python словари – это изменяемые коллекции, которые хранят данные в формате ключ-значение. Каждый элемент словаря состоит из уникального ключа и соответствующего ему значения. Словари в Python являются неупорядоченными, что означает, что элементы в словаре не имеют определенного порядка. Словари в Python представлены фигурными скобками `{}` и могут содержать любые типы данных в качестве значений.

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

Да, функция `len()` может быть использована для определения количества элементов в словаре. Она возвращает количество пар ключ-значение в словаре.

3. Какие методы обхода словарей Вам известны?

Цикл `for-in`: Можно использовать цикл `for-in` для итерации по ключам словаря и получения соответствующих значений. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
```

```
for key in my_dict:
```

```
    value = my_dict[key]
```

```
    print(key, value)
```

Методы `keys()`, `values()` и `items()`: Метод `keys()` возвращает итерируемый объект, содержащий все ключи словаря. Метод `values()` возвращает итерируемый объект, содержащий все значения словаря. Метод `items()` возвращает итерируемый объект, содержащий кортежи (ключ, значение) для каждой пары ключ-значение в словаре. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
```

```
for key in my_dict.keys():
```

```
    print(key)
```

```
for value in my_dict.values():
```

```
    print(value)
```

```
for key, value in my_dict.items():
```

```
    print(key, value)
```

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

В Python можно получить значения из словаря по ключу с помощью следующих способов:

Использование оператора доступа к элементам словаря (`[]`): Можно использовать оператор доступа к элементам словаря, указав ключ в квадратных скобках. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
```

```
print(my_dict['apple'])
```

Метод `get()`: Метод `get()` позволяет получить значение по ключу. Если ключ не существует, метод возвращает значение по умолчанию (`None`, если не указано другое значение). Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
print(my_dict.get('apple'))
```

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

В Python можно установить значение в словаре по ключу с помощью следующих способов:

Использование оператора доступа к элементам словаря (`[]`): Можно использовать оператор доступа к элементам словаря, указав ключ в квадратных скобках и присвоив новое значение. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
my_dict['apple'] = 4
print(my_dict)
```

Метод `update()`: Метод `update()` позволяет обновить словарь, добавив новую пару ключ-значение или обновив значение существующего ключа. Пример:

```
my_dict = {'apple': 3, 'banana': 5, 'orange': 2}
my_dict.update({'apple': 4, 'mango': 6})
print(my_dict)
```

6. Что такое словарь включений?

Словарь включений (`dictionary comprehension`) - это способ создания словаря на основе итерации и условий. Он позволяет создавать словари более компактно и эффективно. Синтаксис словаря включений выглядит следующим образом:

```
{ключ_выражение: значение_выражение for элемент in
итерируемый_объект if условие}
```

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

Функция `zip()` в Python используется для объединения элементов из нескольких итерируемых объектов в один итерируемый объект, состоящий из кортежей. Количество элементов в результирующем итерируемом объекте равно количеству элементов в самом коротком итерируемом объекте.

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

Модуль `datetime` в Python предоставляет функционал для работы с датой и временем. Некоторые возможности модуля `datetime` включают: Создание объектов даты и времени: Модуль `datetime` позволяет создавать объекты для представления даты, времени или комбинации даты и времени.