

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

Институт цифрового разв  
ития Кафедра  
инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

**дисциплины «Основы кроссплатформенного программирования  
»**

Выполнил:  
Плещенко Данила Георгиевич  
1 курс, группа ИТС-б-о-21-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

\_\_\_\_\_  
(подпись)

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

\_\_\_\_\_  
(подпись)

Отчет защищен оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

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

**Ход работы:**

**Пример**

```
primer1.py > ...
73         count += 1
74         print(
75             '{:>4}: {}'.format(count, worker.get('name', ''))
76         )
77         # Если счетчик равен 0, то работники не найдены.
78         if count == 0:
79             print("Работники с заданным стажем не найдены.")
80     elif command == 'help':
81         # Вывести справку о работе с программой.
82         print("Список команд:\n")
83         print("add - добавить работника;")
84         print("list - вывести список работников;")
85         print("select <стаж> - запросить работников со стажем;")
86         print("help - отобразить справку;")
87         print("exit - завершить работу с программой.")
88     else:
89         print(f"Неизвестная команда {command}", file=sys.stderr)
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

```
sktop/учеба/крос/7.lab.rab/primer1.py
>>> add
Фамилия и инициалы? Плещенко д г
Должность? Студент
Год поступления? 2021
>>> list
```

No	Ф.И.О.	Должность	Год
1	Плещенко д г	Студент	2021

Рисунок 1. Работа программы «Пример»

**Задание**

```
individual.py > ...
247 logging.info(f'Сохранены данные в файл {paths[1]}.')
248
249 elif command == 'help':
250     # Вывести справку о работе с программой.
251     print("Список команд:\n")
252     print("add - добавить продукт;")
253     print("list - вывести список продуктов;")
254     print("load <имя_файла> - загрузить данные из файла;")
255     print("save <имя_файла> - сохранить данные в файл;")
256     print("select <товар> - информация о товаре;")
257     print("help - отобразить справку;")
258     print("exit - завершить работу с программой.")
259
260 else:
261     raise UnknownCommandError(command)
262 except Exception as exc:
263     logging.error(f"Ошибка: {exc}")
264     print(exc, file=sys.stderr)
```

ПРОБЛЕМЫ    ВЫХОДНЫЕ ДАННЫЕ    КОНСОЛЬ ОТЛАДКИ    ТЕРМИНАЛ

```
\n -> Unknown command
>>> add
Название товара? колбаса
Название магазина? дубки
Стоимость товара в руб.? 200
>>> list
```

No	Продукт	Магазин	Цена
1	колбаса	дубки	200

Рисунок 2. Работа программы «Индивидуальное задание»

**Вывод:** Я приобрёл 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?

Ответ: Словарь (dict) представляет собой структуру данных (которая ещё называется ассоциативный массив), предназначенную для хранения произвольных объектов с доступом по ключу.

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

Ответ: Да может! Функция `len()` возвращает длину (количество элементов) в объекте.

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

Ответ: У словаря как класса есть метод `items()`, который создает особую структуру, состоящую из кортежей. Каждый кортеж включает ключ и значение:

```
>>> n = nums.items()
>>> n
dict_items([(1, 'one'), (2, 'two'), (3, 'three')])
```

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:

```
>>> v_nums = []
>>> for v in nums.values():
...     v_nums.append(v)
...
>>> v_nums
['one', 'two', 'three']
```

Так существуют методы `clear()`, `copy()`, `fromkeys()`, `get()`, `pop()`, `popitem()`, `setdefault()`, `update()`.

Метод `clear()` удаляет все элементы словаря, но не удаляет сам словарь. В итоге остается пустой Словарь. Метод `fromkeys()` позволяет создать словарь из списка, элементы которого становятся ключами. Применять метод можно как классу `dict`, так и к его объектам. Метод `get()` позволяет получить элемент по его ключу. Метод `pop()` удаляет из словаря элемент по указанному ключу и возвращает значение удаленной пары. Метод `popitem()` не принимает аргументов, удаляет и возвращает произвольный элемент. С помощью `setdefault()` можно добавить элемент в словарь. С помощью `update()` можно добавить в словарь другой словарь

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

Ответ:Операция `dict[key]` вернет элемент словаря `dict` с ключом `key`. Операция вызывает исключение `KeyError`, если ключ `key` отсутствует в словаре.

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

Ответ:Операция `d[key] = value` добавит в словарь `dict` новый элемент - пару ключ-значение.

Если в словаре существует ключ `key` то эта операция присвоит ключу `key` новое значение `value`.

6. Что такое словарь включений?

Ответ:Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка. Как и в случае со списком, мы можем использовать условный оператор внутри словаря включения, чтобы получить только элементы словаря, удовлетворяющие заданному критерию.

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

Ответ:Функция `zip()` создает итератор кортежей, который объединяет элементы каждой из переданных последовательностей `*iterables`.

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

Ответ:`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:

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