

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций  
«Работа со словарями в языке Python»**

**Отчет по лабораторной работе № 2.6  
по дисциплине «Программирование на Python»**

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«6 » сентября 2022г.

Подпись студента \_\_\_\_\_

Работа защищена «    » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2022

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

**Порядок выполнения работы:**

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

Owner \* dshayderov / Repository name \* lw\_2.6

Great repository names are short and memorable. Need inspiration? How about [urban-broccoli](#)?

Description (optional)

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python

Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

Рисунок 1 - Создание репозитория

2. Выполните клонирование созданного репозитория.

```
Asus@LAPTOP-09A994CG MINGW64 ~  
$ cd "C:\My projects\3"  
  
Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3  
$ git clone https://github.com/dshayderov/lw_2.6.git  
Cloning into 'lw_2.6'...  
remote: Enumerating objects: 5, done.  
remote: Counting objects: 100% (5/5), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (5/5), done.
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

```

Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3
$ cd "C:\My projects\3\lw_2.6"

Asus@LAPTOP-09A994CG MINGW64 /c/My projects/3/lw_2.6 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

```

Рисунок 3 - Ветвление по модели git-flow

4. Создайте проект PyCharm в папке репозитория.

Этот компьютер > OS (C:) > My projects > 2 > lw8 >

Имя	Дата изменения	Тип	Размер
Project	30.05.2022 14:54	Папка с файлами	
.gitignore	30.05.2022 14:52	Файл "GITIGNORE"	2 КБ
LICENSE	30.05.2022 14:52	Файл	2 КБ
README.md	30.05.2022 14:52	Файл "MD"	1 КБ

Рисунок 4 - Проект PyCharm

5. Проработайте пример лабораторной работы. Создайте для него отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

```

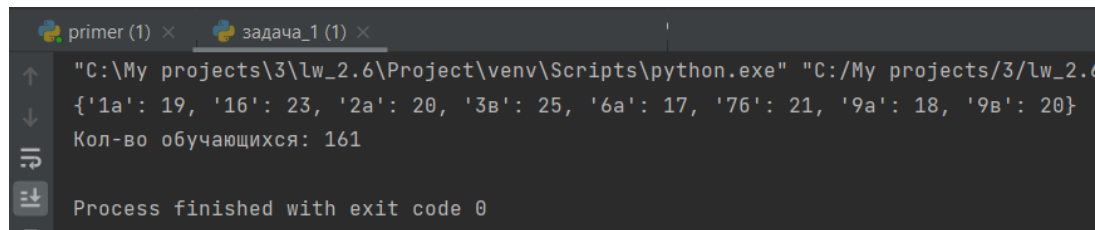
primer (1) x
"C:\My projects\3\lw_2.6\Project\venv\Scripts\python.exe" "C:/My projects/3/lw_
>>> add
Фамилия и инициалы? Петров А.И.
Должность? Менеджер
Год поступления? 2015
>>> add
Фамилия и инициалы? Иванов Л.В.
Должность? Директор
Год поступления? 2011
>>> list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Иванов Л.В.              | Директор             | 2011          |
|  2 | Петров А.И.              | Менеджер              | 2015          |
+-----+-----+-----+-----+
>>> select 10
1: Иванов Л.В.
>>> |

```

Рисунок 5 - Результат выполнения примера

6. 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

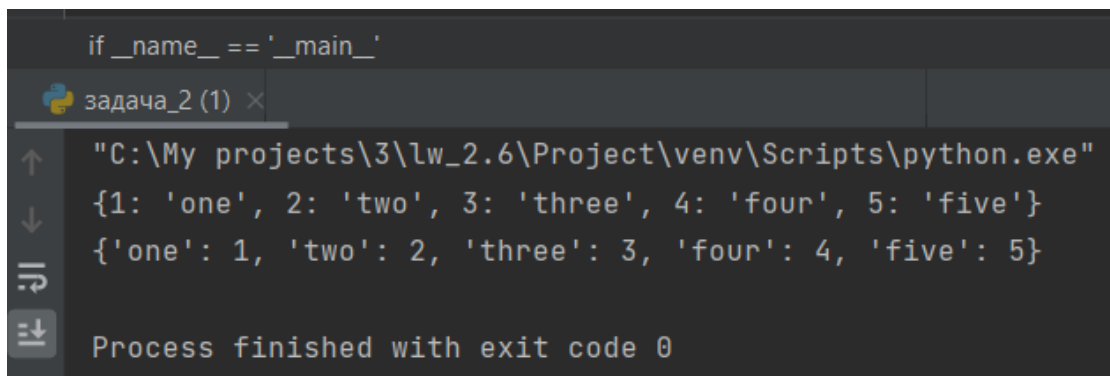
школе появился новый класс, с) в школе был расформирован (удален) другой класс. Вычислите общее количество учащихся в школе.



```
primer (1) x задача_1 (1) x
"C:\My projects\3\lw_2.6\Project\venv\Scripts\python.exe" "C:/My projects/3/lw_2.6
{'1a': 19, '1b': 23, '2a': 20, '3b': 25, '6a': 17, '7b': 21, '9a': 18, '9b': 20}
Кол-во обучающихся: 161
Process finished with exit code 0
```

Рисунок 6 - Результат выполнения задачи 1

7. 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.



```
задача_2 (1) x
"C:\My projects\3\lw_2.6\Project\venv\Scripts\python.exe"
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}
Process finished with exit code 0
```

Рисунок 7 - Результат выполнения задачи 2

8. Выполните индивидуальное задание (Вариант 24/5).

Использовать словарь, содержащий следующие ключи: название пункта назначения рейса; номер рейса; тип самолета. 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 пунктов назначения; вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

```
w_1 x
"C:\My projects\3\lw_2.6\Project\venv\Scripts\python.exe" "C:\My projects\3\lw_2.6\Pro
>>> add
Пункт назначения: Москва
Номер рейса: 123
Тип самолета: грузовой
>>> add
Пункт назначения: Санкт-Петербург
Номер рейса: 456
Тип самолета: пассажирский
>>> list
+-----+-----+-----+-----+
| No | Пункт назначения | Номер рейса | Тип самолета |
+-----+-----+-----+-----+
| 1 | Москва | 123 | грузовой |
| 2 | Санкт-Петербург | 456 | пассажирский |
+-----+-----+-----+-----+
>>> select пассажирский
1. Пункт: Санкт-Петербург; Номер: 456
>>> exit
```

Рисунок 8 - Результат выполнения индивидуального задания

## Контрольные вопросы

### 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?

Да, функция len() определяет длину словаря.

### 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. Но по ключам всегда можно получить значения.

С другой стороны у словаря как класса есть метод items(), который создает особую структуру, состоящую из кортежей. Каждый кортеж включает ключ и значение. В цикле for можно распаковывать кортежи, таким образом сразу извлекая как ключ, так и его 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?**

Обращением по ключу в квадратных скобках (`dict[key]`) или при помощи метода `get()`.

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

Обращением по ключу в квадратных скобках (`dict[key]`) или при помощи метода `setdefault()`, если элемент с указанным ключом отсутствует в списке.

**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.

В Python есть несколько встроенных функций, которые позволяют перебирать данные. Одна из них — `zip`. Функция `zip()` в Python создает итератор, который объединяет элементы из нескольких источников данных.

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.

**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.

Класс `datetime.date(year, month, day)` - стандартная дата. Атрибуты: `year, month, day`. Неизменяемый объект.

Класс `datetime.time(hour=0, minute=0, second=0, microsecond=0, tzinfo=None)` - стандартное время, не зависит от даты. Атрибуты: `hour, minute, second, microsecond, tzinfo`.

Класс `datetime.timedelta` - разница между двумя моментами времени, с точностью до микросекунд.

Класс `datetime.tzinfo` - абстрактный базовый класс для информации о временной зоне (например, для учета часового пояса и / или летнего времени).

Класс `datetime.datetime(year, month, day, hour=0, minute=0, second=0, microsecond=0, tzinfo=None)` - комбинация даты и времени.

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