

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

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №1**

**Работа со словарями в языке Python**

**По дисциплине «Теории программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

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

Проверил Воронкин Р. А. \_\_\_\_\_

(подпись)

Ставрополь 2021

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

### Ход работы

1. Создал новый собственный репозиторий. Ссылка на репозиторий: [https://github.com/Dmitry-15/8\\_laba.git](https://github.com/Dmitry-15/8_laba.git).
2. С помощью команды `git clone` клонировал удаленный репозиторий на свой ПК. Дополнил файл `.gitignore` необходимыми правилами для работы с IDE PyCharm.
3. Проработал пример лабораторной работы.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from datetime import date
import sys

if __name__ == '__main__':
    # Список работников.
    workers = []

    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()

        # Выполнить действие в соответствие с командой.
        if command == 'exit':
            break

        elif command == 'add':
            # Запросить данные о работнике.
            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
            year = int(input("Год поступления? "))
            # Создать словарь.
            worker = {
                'name': name,
                'post': post,
                'year': year,
            }
```

Рисунок 1. Код примера

```

C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/8_laba/primer1.py
>>> add
Фамилия и инициалы? Плотников Д. В.
Должность? Директор
Год поступления? 2018
>>> select 2
1: Плотников Д. В.
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Плотников Д. В. | Директор | 2018 |
+-----+-----+-----+-----+

```

Рисунок 2. Выполнение примера

### Задание 1

1. Условие задачи: 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.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':

    school = {}
    for i in range(7):
        school.update(
            {input(f'Название класса № {i + 1} : '): int(input(f'Количество учеников в классе № {i + 1} : '))})
    print(school)
    klas = input('Введите название класса: ')
    if klas in school:
        print(f'Количество учеников в классе {klas} : {school[klas]} ')
    else:
        print('Такого класса не существует')
    for i in range(2):
        school.update({input(f'Название изменяемого класса {i + 1} : '): int(
            input(f'Количество учеников изменяемого класса {i + 1} : '))})
    print(school)
    for i in range(1):
        school.update(
            {input(f'Название нового класса {i + 1} : '): int(input(f'Количество учеников нового класса {i + 1} : '))})
    print(school)
    del school[input(f'Название расформировываемого класса: ')]
    print(school)

```

Рисунок 3. Код задания 1

```

C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/8_laba/zadanie1.py
Название класса № 1 : а
Количество учеников в классе № 1 : 20
Название класса № 2 : б
Количество учеников в классе № 2 : 21
Название класса № 3 : в
Количество учеников в классе № 3 : 22
Название класса № 4 : г
Количество учеников в классе № 4 : 23
Название класса № 5 : д
Количество учеников в классе № 5 : 24
Название класса № 6 : е
Количество учеников в классе № 6 : 25
Название класса № 7 : ё
Количество учеников в классе № 7 : 26
{'a': 20, 'б': 21, 'в': 22, 'г': 23, 'д': 24, 'е': 25, 'ё': 26}
Введите название класса: д
Количество учеников в классе д : 24
Название изменяемого класса 1 : б
Количество учеников изменяемого класса 1 : 25
Название изменяемого класса 2 : а
Количество учеников изменяемого класса 2 : 23
{'a': 23, 'б': 25, 'в': 22, 'г': 23, 'д': 24, 'е': 25, 'ё': 26}
Название нового класса 1 : и
Количество учеников нового класса 1 : 30
{'a': 23, 'б': 25, 'в': 22, 'г': 23, 'д': 24, 'е': 25, 'ё': 26, 'и': 30}
Название расформировываемого класса: в
{'a': 23, 'б': 25, 'в': 22, 'г': 23, 'д': 24, 'е': 25, 'и': 30}

```

Рисунок 4. Выполнение задания 1

## Задание 2

1. Условие задачи: 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.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

words = {
    1: 'one',
    2: 'two',
    3: 'three',
    4: 'four'
}

dict_items = words.items()
new_words = dict(zip(words.values(), words.keys()))
print(new_words)

```

Рисунок 5. Код задания 2

```
C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/8_laba/zadanie2.py
{'one': 1, 'two': 2, 'three': 3, 'four': 4}

Process finished with exit code 0
```

Рисунок 6. Выполнение задания 2

## Индивидуальное задание

### Вариант 14

1. Условие задачи: Использовать словарь, содержащий следующие ключи: фамилия, имя; знак Зодиака; дата рождения (список из трех чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по датам рождения; вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    workers = []
    while True:
        command = input(">>> ").lower()
        if command == 'exit':
            break
        elif command == 'add':
            name = input("Фамилия и инициалы? ")
            zodiac = input("Знак Зодиака? ")
            year = list(map(int, input("Дата рождения? ").split()))
            worker = {
                'name': name,
                'zodiac': zodiac,
                'year': year,
            }
            workers.append(worker)
            if len(workers) > 1:
                workers.sort(key=lambda x: x.get('year')[::-1])
        elif command == 'list':
            line = '+-{}-+-{}-+-{}-+-{}-+'.format(
                '-' * 4,
                '-' * 30,
                '-' * 20,
                '-' * 15
            )
            print(line)
            print(
```

Рисунок 7. Код индивид. Задания

```

C:\ProgramData\Anaconda3\python.exe C:/Users/Дмитрий/Pictures/8_laba/individ.py
>>> add
Фамилия и инициалы? Плотников Д.В.
Знак Зодиака? лев
Дата рождения? 15 8 2002
>>> add
Фамилия и инициалы? Бобров Н.В.
Знак Зодиака? стрелец
Дата рождения? 19 12 2002
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Плотников Д.В. | лев | 15 8 2002 |
| 2 | Бобров Н.В. | стрелец | 19 12 2002 |
+-----+-----+-----+-----+
>>> whois
Кого ищем?: Плотников Д.В.
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
| 1 | Плотников Д.В. | лев | 15 8 2002 |
+-----+-----+-----+-----+
>>> whois
Кого ищем?: Галакс
+-----+-----+-----+-----+
| № | Ф.И.О. | Знак Зодиака | Дата рождения |
+-----+-----+-----+-----+
Никто не найден

```

Рисунок 8. Выполнение индивид. задания

## Контрольные вопросы

1. Что такое словари в языке Python?

Словарь – это структура данных, которая предназначена для хранения произвольных объектов с доступом по ключу.

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

Да, может.

3. Какие методы обхода словарей Вам известны?

- For I in dict
- for key in dict.keys()
- for value in dict.value()
- for key, value in dict.items()

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

Print(dict[‘Название ключа’])

Print(dict.get(‘Название ключа’))

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

`Dict['Ключ'] = значение`

6. Что такое словарь исключений?

Исключение в Python – это конструкция, используемая для сигнализации о важном событии, обычно об ошибке, которое происходит при выполнении программы. Исключение может привести к остановке программы, если она не будет должным образом «поймана» (т.е. обработана правильно). Если вы думаете, что ваша программа может вызвать исключение при выполнении.

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

Функция `zip` объединяет в кортежи элементы из последовательностей переданных в качестве аргументов. Функция прекращает выполнение, как только достигнут конец самого короткого списка.

```
a = [1,2,3]
b = "xyz"
c = (None, True)
res = list(zip(a, b, c))
print (res)
[(1, 'x', None), (2, 'y', True)]
```

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.

Обязательные аргументы:

- `datetime.MINYEAR (1) ≤ year ≤ datetime.MAXYEAR (9999)`
- `1 ≤ month ≤ 12`
- `1 ≤ day ≤ количество дней в данном месяце и году`

Методы класса `datetime`:

– `datetime.today()` - объект `datetime` из текущей даты и времени. Работает также, как и `datetime.now()` со значением `tz=None`.

– `datetime.fromtimestamp(timestamp)` - дата из стандартного представления времени.

– `datetime.fromordinal(ordinal)` - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970.

– `datetime.now(tz=None)` - объект `datetime` из текущей даты и времени.

– `datetime.combine(date, time)` - объект `datetime` из комбинации объектов `date` и `time`.

– `datetime.strptime(date_string, format)` - преобразует строку в `datetime` (так же, как и функция `strptime` из модуля `time`).

– `datetime.strftime(format)` - см. функцию `strftime` из модуля `time`.

– `datetime.date()` - объект даты (с отсечением времени).

– `datetime.time()` - объект времени (с отсечением даты).

– `datetime.replace([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]])` - возвращает новый объект `datetime` с изменёнными атрибутами.

– `datetime.timetuple()` - возвращает `struct_time` из `datetime`.



- `datetime.toordinal()` - количество дней, прошедших с 01.01.1970.
- `datetime.timestamp()` - возвращает время в секундах с начала эпохи.
- `datetime.weekday()` - день недели в виде числа, понедельник - 0, воскресенье - 6.
- `datetime.isoweekday()` - день недели в виде числа, понедельник - 1, воскресенье - 7.
- `datetime.isocalendar()` - кортеж (год в формате ISO, ISO номер недели, ISO день недели).
- `datetime.isoformat(sep='T')` - красивая строка вида "YYYY-MM-DDTHH:MM:SS.mmmmmm" или, если `microsecond == 0`, "YYYY-MM-DDTHH:MM:SS"
- `datetime.ctime()` - см. `ctime()` из модуля `time`.

**Вывод:** в ходе выполнения лабораторной работы успешно приобрел навыки по работе со словарями при написании программ с помощью языка программирования Python3.