

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»**

Выполнил:  
Говоров Егор Юрьевич  
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. технических  
наук, 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.

## Ход работы

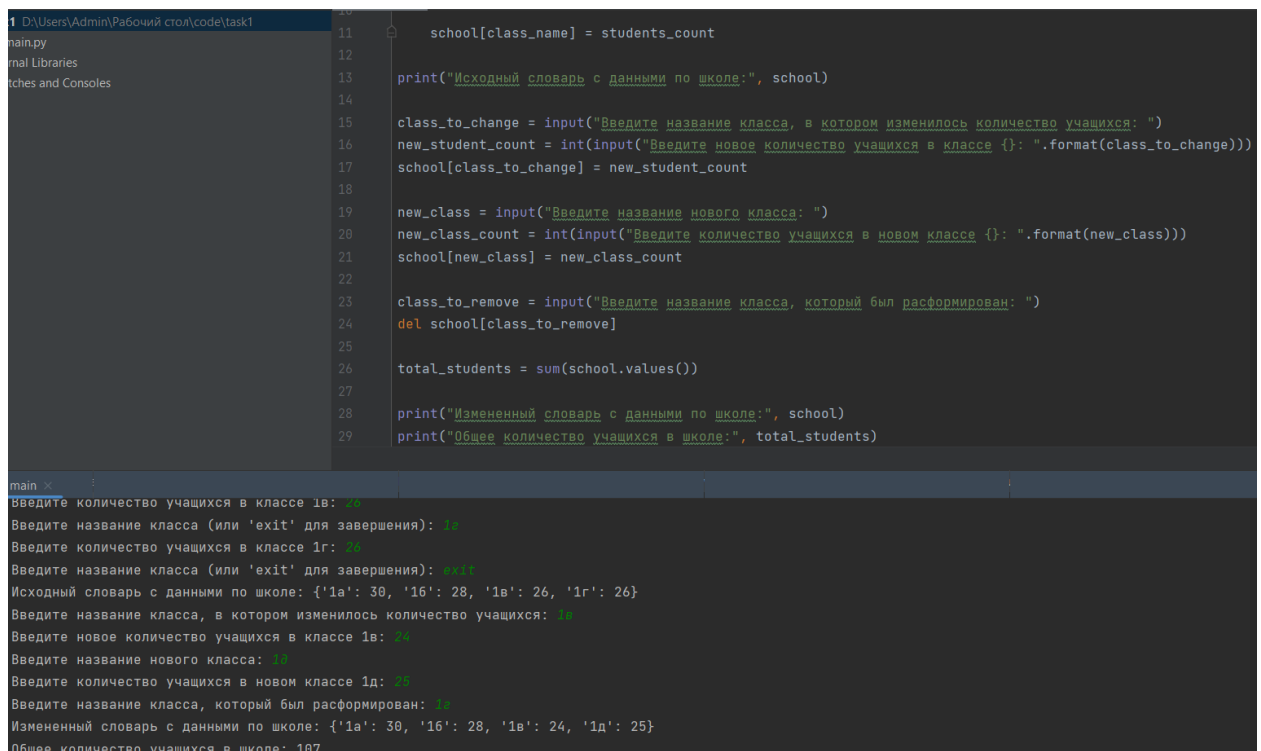
1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.
4. Проработал пример лабораторной работы. Создал для него отдельный модуль языка Python. Привел в отчете скриншоты результата выполнения программы примера при различных исходных данных, вводимых с клавиатуры.

```
ct
mer D:\Users\Admin\Рабочий стол\code\primer
main.py
ernal Libraries
atches and Consoles

1 #!usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 from distutils import command
5 import sys
6 from datetime import date
7
8 if __name__ == '__main__':
9
10     workers = []
11
12     while True:
13         command = input(">>> ").lower()
14
15         if command == 'exit':
16             break
17
18         elif command == 'add':
19             name = input("Фамилия и инициалы? ")
20             post = input("Должность? ")
21             year = int(input("Год поступления? "))
22
23             worker = {
24                 if __name__ == '__main__' > while True > elif command == 'help'
25
26 Год поступления? 2022
27 >>> list
28
29 +-----+
30 | № |          Ф.И.О.          |      Должность      |      Год      |
31 +-----+
32 |  1 | Говоров.Е.Ю             |      Студент       |      2022     |
33 +-----+
34 >>>
```

Рисунок 1. Результат работы программы из примера 1

5. Решил следующую 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.



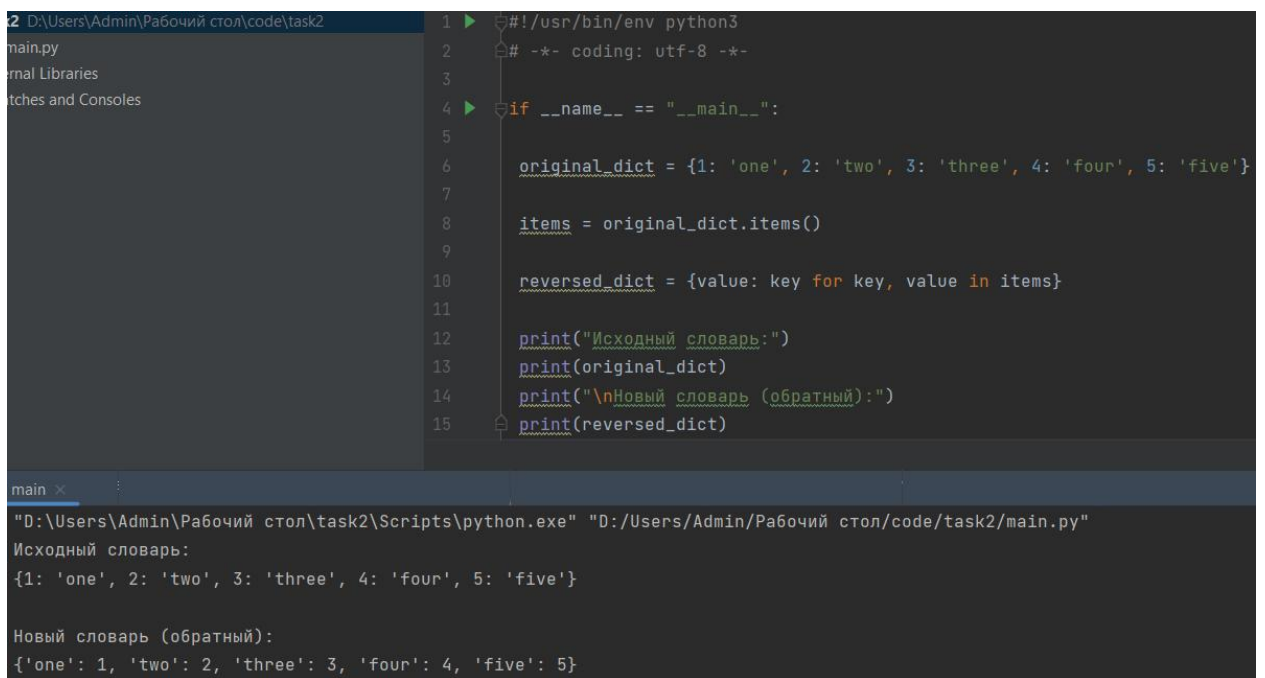
```
1 D:\Users\Admin\Рабочий стол\code\task1
main.py
ernal Libraries
atches and Consoles

10 school[class_name] = students_count
11
12
13 print("Исходный словарь с данными по школе:", school)
14
15 class_to_change = input("Введите название класса, в котором изменилось количество учащихся: ")
16 new_student_count = int(input("Введите новое количество учащихся в классе {}: ".format(class_to_change)))
17 school[class_to_change] = new_student_count
18
19 new_class = input("Введите название нового класса: ")
20 new_class_count = int(input("Введите количество учащихся в новом классе {}: ".format(new_class)))
21 school[new_class] = new_class_count
22
23 class_to_remove = input("Введите название класса, который был расформирован: ")
24 del school[class_to_remove]
25
26 total_students = sum(school.values())
27
28 print("Измененный словарь с данными по школе:", school)
29 print("Общее количество учащихся в школе:", total_students)

main x
Введите количество учащихся в классе 1а: 30
Введите название класса (или 'exit' для завершения): 1а
Введите количество учащихся в классе 1г: 26
Введите название класса (или 'exit' для завершения): exit
Исходный словарь с данными по школе: {'1а': 30, '1б': 28, '1в': 26, '1г': 26}
Введите название класса, в котором изменилось количество учащихся: 1а
Введите новое количество учащихся в классе 1а: 24
Введите название нового класса: 1д
Введите количество учащихся в новом классе 1д: 25
Введите название класса, который был расформирован: 1г
Измененный словарь с данными по школе: {'1а': 30, '1б': 28, '1в': 24, '1д': 25}
Общее количество учащихся в школе: 107
```

Рисунок 2. Результат работы программы из задачи 1

6. Решил следующую 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 if __name__ == "__main__":
5
6     original_dict = {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
7
8     items = original_dict.items()
9
10    reversed_dict = {value: key for key, value in items}
11
12    print("Исходный словарь:")
13    print(original_dict)
14    print("\nНовый словарь (обратный):")
15    print(reversed_dict)
```

main x

"D:\Users\Admin\Рабочий стол\task2\Scripts\python.exe" "D:/Users/Admin/Рабочий стол/code/task2/main.py"

Исходный словарь:  
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}

Новый словарь (обратный):  
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}

Рисунок 3. Результат работы программы из задачи 2

7. Выполнил индивидуальное задание, согласно варианту 5. Привёл в отчете скриншот работы программы.

Задание. 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.





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

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