

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
Институт цифрового развития**

ОТЧЁТ

по лабораторной работе №1

Дисциплина: «Программирование на Python»

Тема: “ Работа со словарями в языке Python ”

Выполнил студент группы

ИВТ-б-о-21-1

Харченко Богдан Романович

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

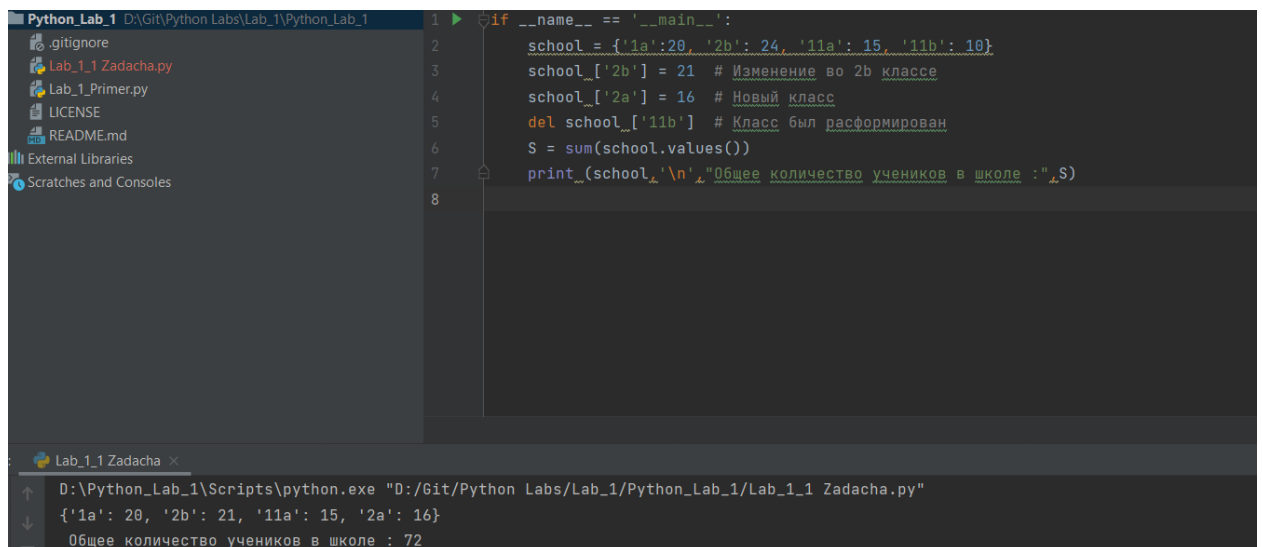
Ставрополь 2022

Ход работы:

```
>>> add
Фамилия и инициалы? Харченко Б.Р.
Должность? Программист
Год поступления? 2021
>>> ist
>>> Неизвестная команда ist
list
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
|  1 | Харченко Б.Р.          | Программист        |      2021     |
+-----+-----+-----+-----+
>>> select
>>> Неизвестная команда select
help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> select 1
      1: Харченко Б.Р.
```

Рисунок 1 – Проработал пример 1 с разными данными



The image shows a Python IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'Python_Lab_1' with files like '.gitignore', 'Lab_1_1 Zadacha.py', 'Lab_1_Primer.py', 'LICENSE', 'README.md', 'External Libraries', and 'Scratches and Consoles'. The code editor shows the following Python code:

```
1 if __name__ == '__main__':
2     school = {'1a': 20, '2b': 24, '11a': 15, '11b': 10}
3     school_['2b'] = 21 # Изменение во 2б классе
4     school_['2a'] = 16 # Новый класс
5     del school_['11b'] # Класс был расформирован
6     S = sum(school.values())
7     print_(school_'\n'_"Общее количество учеников в школе :"_S)
8
```

The execution results are shown in the bottom panel:

```
Lab_1_1 Zadacha x
D:\Python_Lab_1\Scripts\python.exe "D:/Git/Python Labs/Lab_1/Python_Lab_1/Lab_1_1 Zadacha.py"
{'1a': 20, '2b': 21, '11a': 15, '2a': 16}
Общее количество учеников в школе : 72
```

Рисунок 2 – Код для задачи 1 и результаты выполнения программы

```
num = {1: 'a', 2: 'b', 3: 'c'}
print({v:k for k, v in num.items()})
```

Lab_1_2_Zadacha ×

D:\Python_Lab_1\Scripts\python.exe "D:/Git/Python Labs/Lab_1_2_Zadacha.py"

{'a': 1, 'b': 2, 'c': 3}

Process finished with exit code 0

Рисунок 3 – Код для задачи 2 и результаты выполнения программы

```
for i in range(2):
    d = dict()

    d['Фамилия и инициалы'] = input("Введите ФИО: ")
    d['Номер группы'] = int(input("Введите номер группы: "))
    d['Успеваемость'] = list(map(int, input("Введите оценки: ").split()))

    list_of_dicts.append(d)

list_of_dicts = sorted(list_of_dicts, key=lambda x: x["Фамилия и инициалы"])

is_find = False

for item in list_of_dicts:
    if item['Успеваемость'].__contains__(2):
        print(f'Двоишники:\n {item["Фамилия и инициалы"]} \nГруппа: {item["Номер группы"]} \n\n')
        is_find = True

if not is_find:
    print("Нет двоишников!")

if __name__ == '__main__':
    if __name__ == '__main__':
        if not is_find:
```

Lab_1_Individual ×

Введите оценки: 4 5 4 5 5

Введите ФИО: Криворот В.Г.

Введите номер группы: 1

Введите оценки: 2 4 5 5 5

Двоишники:

Криворот В.Г.

Группа: 1

Рисунок 4 – Выполнение индивидуального задания

Контрольные вопросы

1. Что такое словари в языке Python?

Словари в Python — это изменяемые отображения ссылок на объекты, доступные по ключу.

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

Функция `len()` возвращает длину (количество элементов) в объекте. Аргумент может быть последовательностью, такой как строка, байты, кортеж, список или диапазон или коллекцией (такой как **словарь**, множество или неизменяемое множество).

3. Какие методы обхода словарей Вам известны?

Самый очевидный вариант обхода словаря — это попытаться напрямую запустить цикл **for** по объекту словаря, так же как мы делаем это со списками, кортежами, строками и любыми другими итерируемыми объектами.

```
for something in currencies:
```

```
    print(something)
```

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

С помощью метода `.get()`

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

С помощью функции `dict.update()`

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.

Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию **`zip()`**. Вот пример программы, которая делает именно это:

```
employee_numbers = [2, 9, 18, 28]

employee_names = ["Дима", "Марина", "Андрей", "Никита"]

zipped_values = zip(employee_names, employee_numbers)

zipped_list = list(zipped_values)

print(zipped_list)
```

Функция `zip` возвращает следующее:

```
[('Дима', 2), ('Марина', 9), ('Андрей', 18), ('Никита', 28)]
```

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

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. Так, он состоит из объектов следующих типов:

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

Как получить текущие дату и время?

```
import datetime

dt_now = datetime.datetime.now()

print(dt_now)
```

А вот результат:

2022-09-11 15:43:32.249588

Получить текущую дату:

```
from datetime import date

current_date = date.today()

print(current_date)
```

Результат:

2022-09-11

Получить текущее время:

```
import datetime

current_date_time = datetime.datetime.now()

current_time = current_date_time.time()

print(current_time)
```

Результат:

15:51:05.627643

