

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 РАБОТЕ №2.6**  
**дисциплины «Основы кроссплатформенного программирования»**

Выполнил:  
Кондратенко Даниил Витальевич  
1 курс, группа ИТС-б-о-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А., канд. тех. наук, доцент,  
доцент кафедры инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

*Тема:* работа со словарями в языке Python.

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

Порядок выполнения работы:

Задание 1.

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

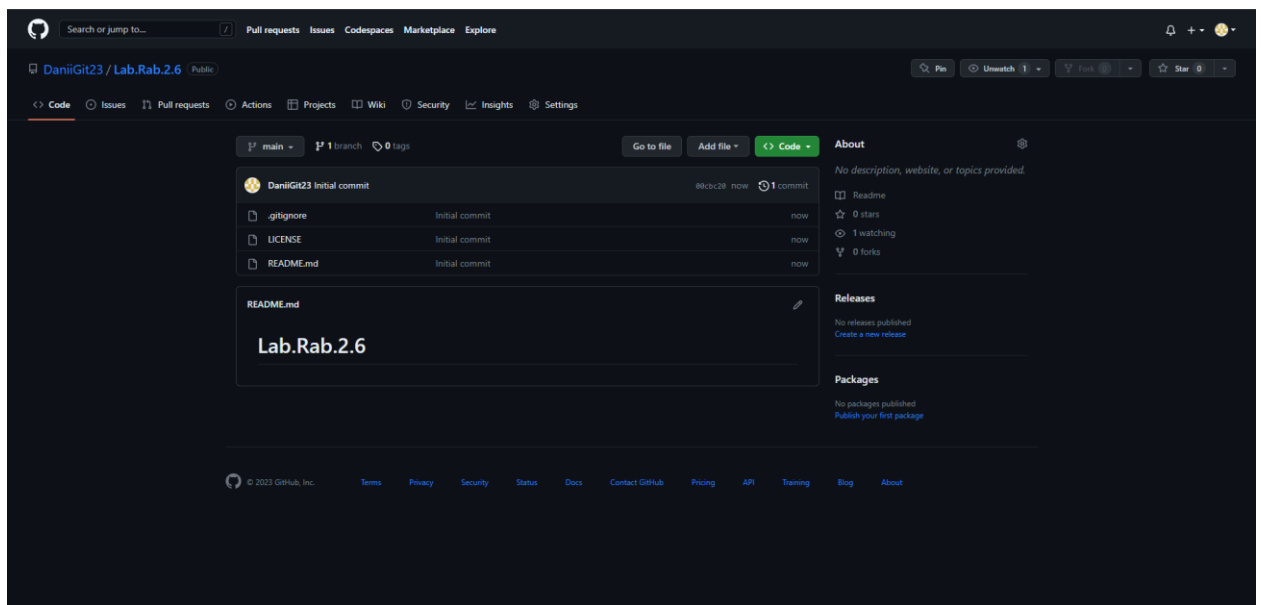


Рисунок 1 – Новый репозиторий

## Задание 2.

Проклонировал свой репозиторий на свой компьютер.

Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop.

```
C:\Users\HUAWEI>git config --global user.name "Daniil"

C:\Users\HUAWEI>git config --global user.email "kondratenko_danil.23@mail.ru"

C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.6.git
Cloning into 'Lab.Rab.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.

C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.6

C:\Users\HUAWEI\Lab.Rab.2.6>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] t
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.6/.git/hooks]
```

Рисунок 2 – Клонирование и модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop, без создания дополнительной ветки feature/(название ветки) по указанию преподавателя.

## Задание 3.

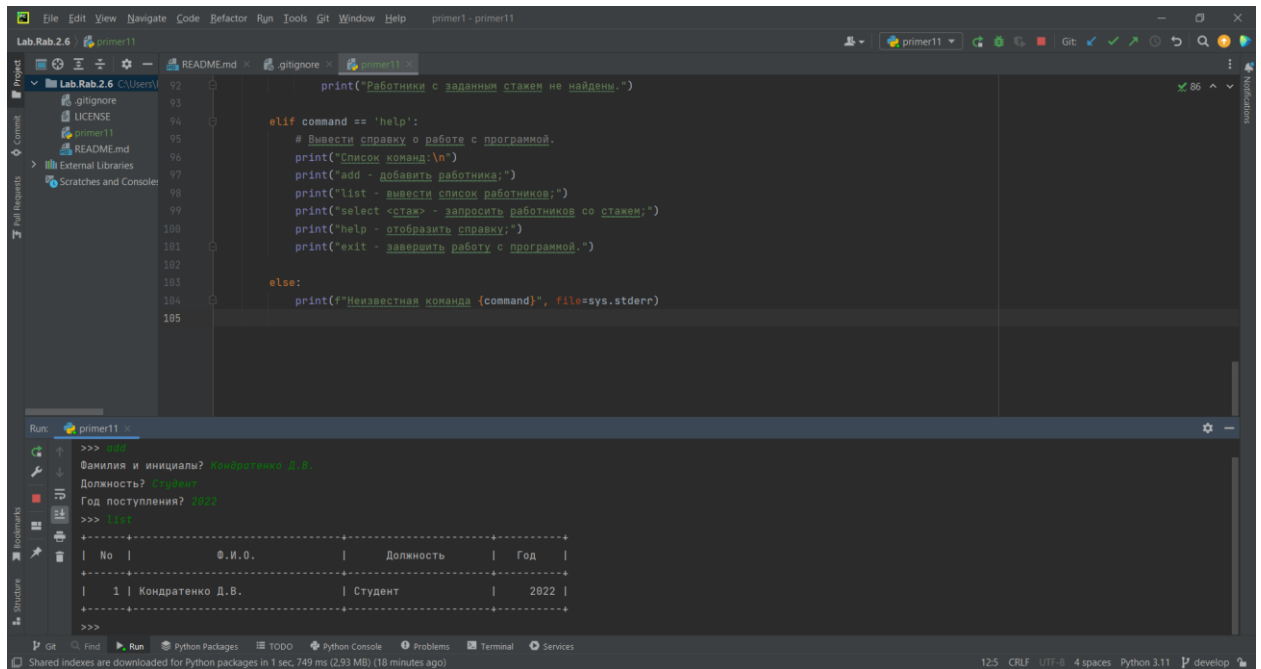
Создал проект PyCharm в папке репозитория.

Работа с примером №1.

Добавил новый файл *primer11.py*.

*Условие примера:* Использовать словарь, содержащий следующие ключи: фамилия и инициалы работника; название занимаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия:

- 1) ввод с клавиатуры данных в список, состоящий из заданных словарей;
- 2) записи должны быть размещены по алфавиту;
- 3) вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
- 4) если таких работников нет, вывести на дисплей соответствующее сообщение.



The screenshot shows a code editor with a Python script and a terminal window. The script defines a menu-driven program for managing employee data. The terminal shows the program's execution, where user input is used to add a new employee, and the resulting data is displayed in a formatted table.

```
print("Работники с заданным стажем не найдены.")

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить работника;")
    print("list - вывести список работников;")
    print("select <стаж> - запросить работников со стажем;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")
else:
    print(f"Неизвестная команда {command}", file=sys.stderr)
```

Run: primer11

```
>>>
Фамилия и инициалы? Кондратенко Д.В.
Должность? Студент
Год поступления? 2022
>>> list
-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
-----+-----+-----+-----+
| 1 | Кондратенко Д.В. | Студент | 2022 |
-----+-----+-----+-----+
>>>
```

Рисунок 3 – Реализация примера 1

#### Задание 4.

Создал новый файл *ObsheeZadanye1.py*

*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.

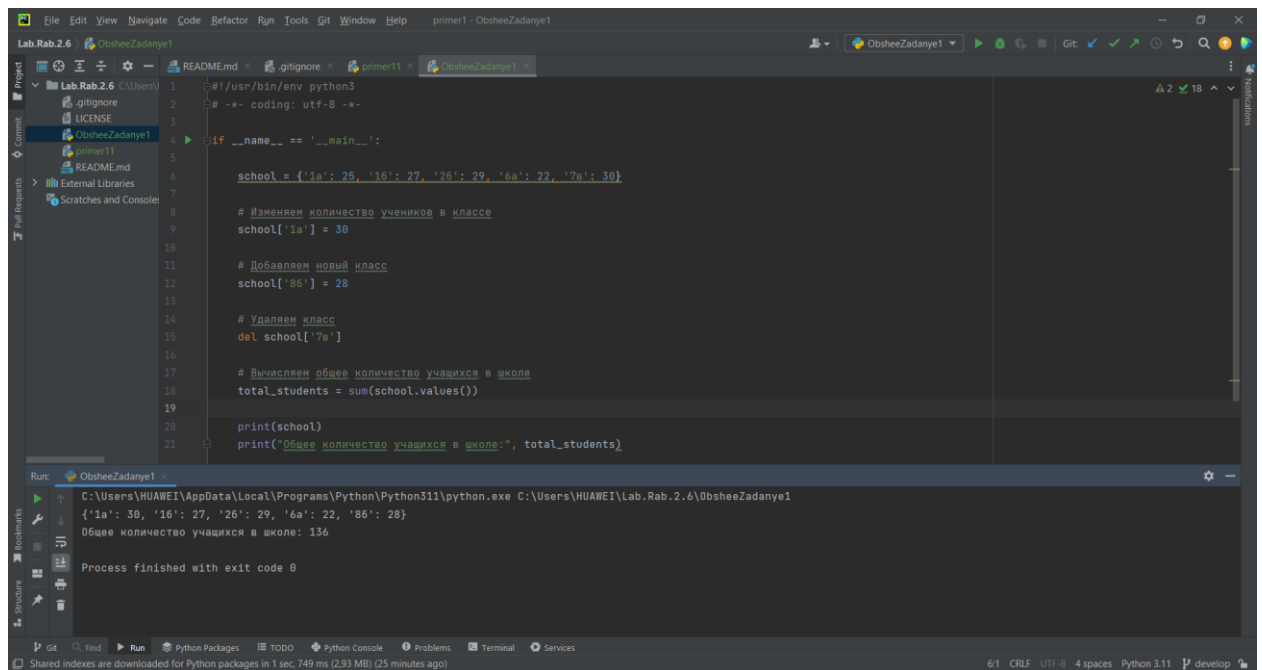


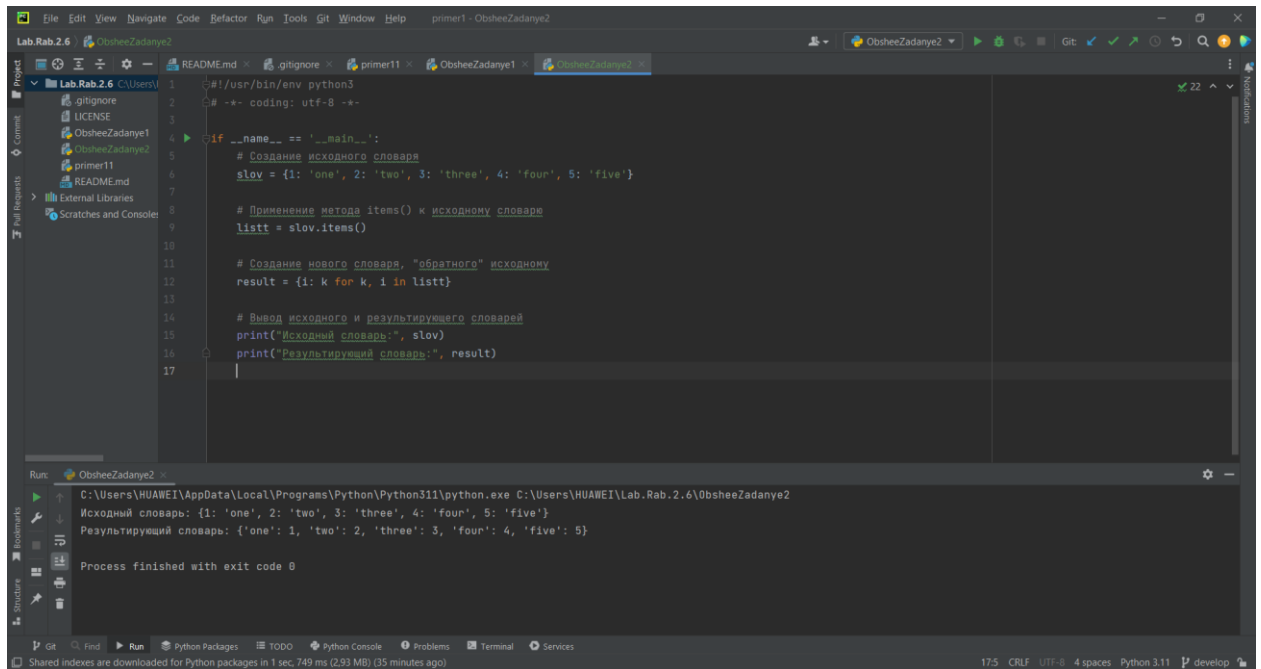
Рисунок 4 – Реализация общего задания 1

Зафиксировал данные изменения в репозитории.

## Задание 5.

Создал новый файл *ObsheeZadany2.py*

*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.



The screenshot shows a Python IDE with a file named `ObsheeZadany2.py`. The code defines a dictionary `slov` with keys 1-5 and values 'one'-'five'. It uses `slov.items()` to create a list of items, then constructs a new dictionary `result` where the keys and values are swapped. The output shows the original and inverted dictionaries.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     # Создание исходного словаря
6     slov = {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
7
8     # Применение метода items() к исходному словарю
9     listt = slov.items()
10
11     # Создание нового словаря, "обратного" исходному
12     result = {i: k for k, i in listt}
13
14     # Вывод исходного и результирующего словарей
15     print("Исходный словарь:", slov)
16     print("Результирующий словарь:", result)
17
```

Run: C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI\Lab.Rab.2.6\ObsheeZadany2

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

Process finished with exit code 0

Рисунок 5 – Реализация общего задания 2

## Задание 6.

Выполнение индивидуального задания.

Создал новый файл под названием *individual.py*.

*Условие задания:* Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть размещены по алфавиту; вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры; если таких нет, выдать на дисплей соответствующее сообщение.

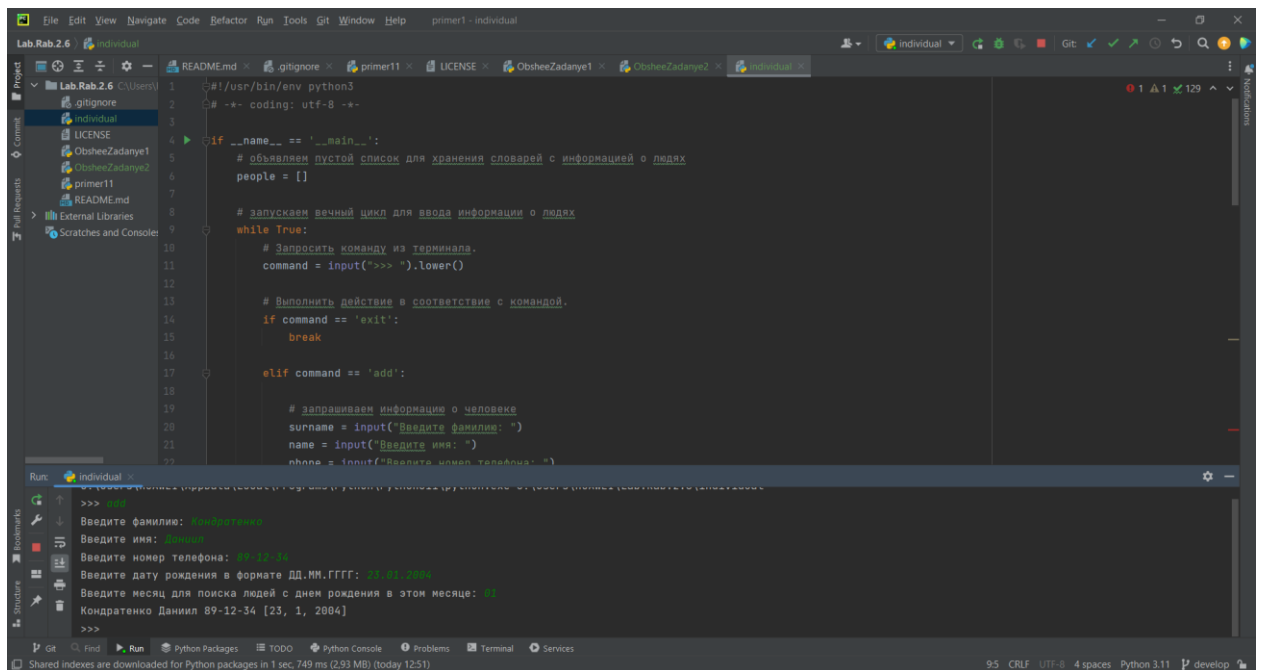


Рисунок 6 – Программа и ее результат

Ссылка на репозиторий: <https://github.com/DaniiGit23/Lab.Rab.2.6.git>

Контрольные вопросы:

1. Что такое словари в языке Python?

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?

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

3. Какие методы обхода словарей Вам известны?

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 же 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?

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?

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?

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.

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?

`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

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