

**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 «Технологии программирования и алгоритмизация»**

Выполнил студент группы ИВТ-б-о-19-1

Гончаров Даниил Ростиславович.

«09» сентября 2021г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Преподаватель 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:

Ответы на 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?

Может, как и в случае со списком, мы можем использовать условный оператор внутри словаря включения, чтобы получить только элементы словаря, удовлетворяющие заданному критерию.

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.

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

Метод get() позволяет получить элемент по его ключу.

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

С помощью setdefault() можно добавить элемент в словарь.

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.

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.

Решение заданий по лабораторной работе

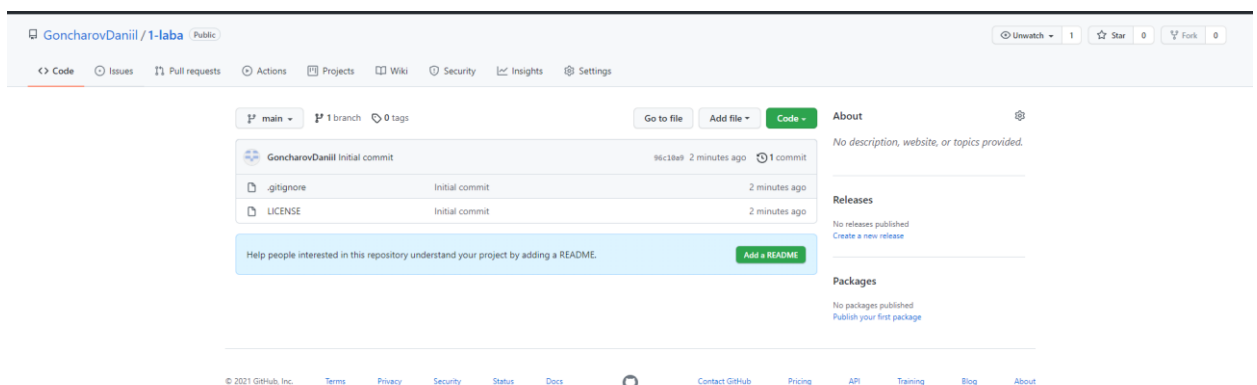


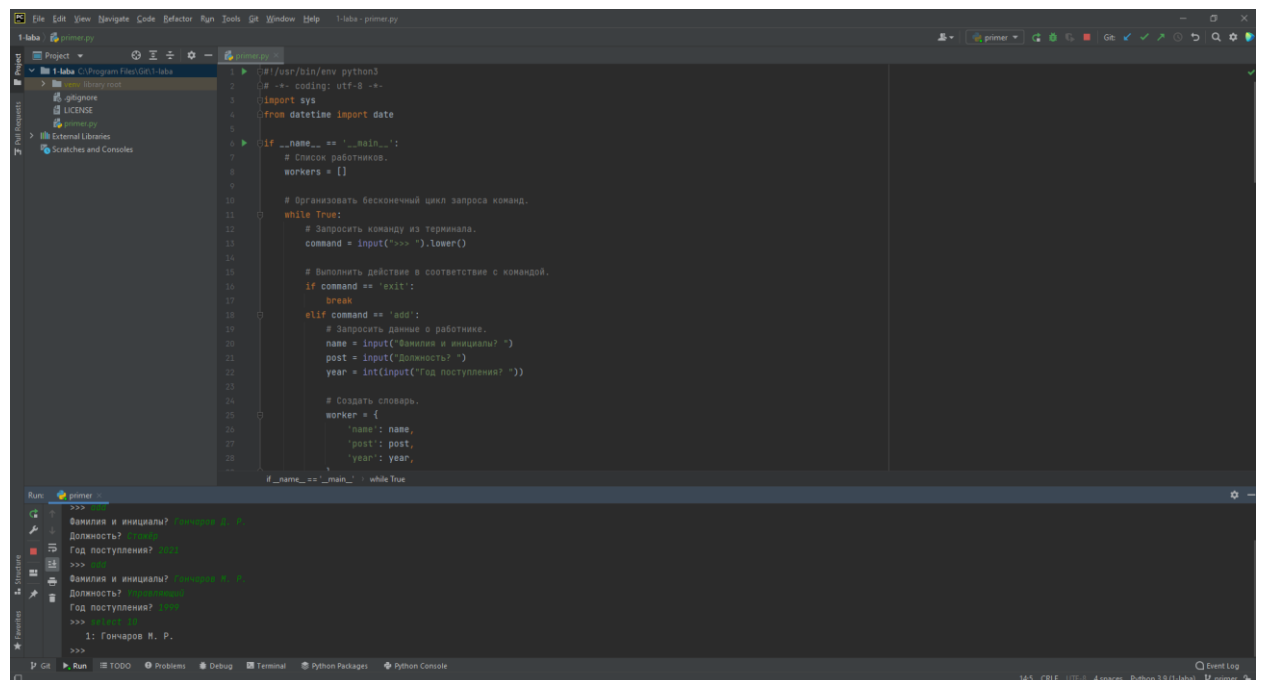
Рисунок 1. Создание репозитория

```
C:\Program Files\Git>git clone https://github.com/GoncharovDaniil/1-laba.git
Cloning into '1-laba'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 2.17 KiB | 148.00 KiB/s, done.

C:\Program Files\Git>
```

Рисунок 2. Копирование репозитория на ПК

Задание 1. Пример лабораторной работы.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 import sys
4 from datetime import date
5
6 if __name__ == '__main__':
7     # Список работников.
8     workers = []
9
10    # Организовать бесконечный цикл запроса команд.
11    while True:
12        # Запросить команду из терминала.
13        command = input(">>> ").lower()
14
15        # Выполнить действие в соответствие с командой.
16        if command == 'exit':
17            break
18        elif command == 'add':
19            # Запросить данные о работнике.
20            name = input("Фамилия и инициалы? ")
21            post = input("Должность? ")
22            year = int(input("Год поступления? "))
23
24            # Создать словарь.
25            worker = {
26                'name': name,
27                'post': post,
28                'year': year,
29            }
30            workers.append(worker)
31
32    if __name__ == '__main__': while True:
```

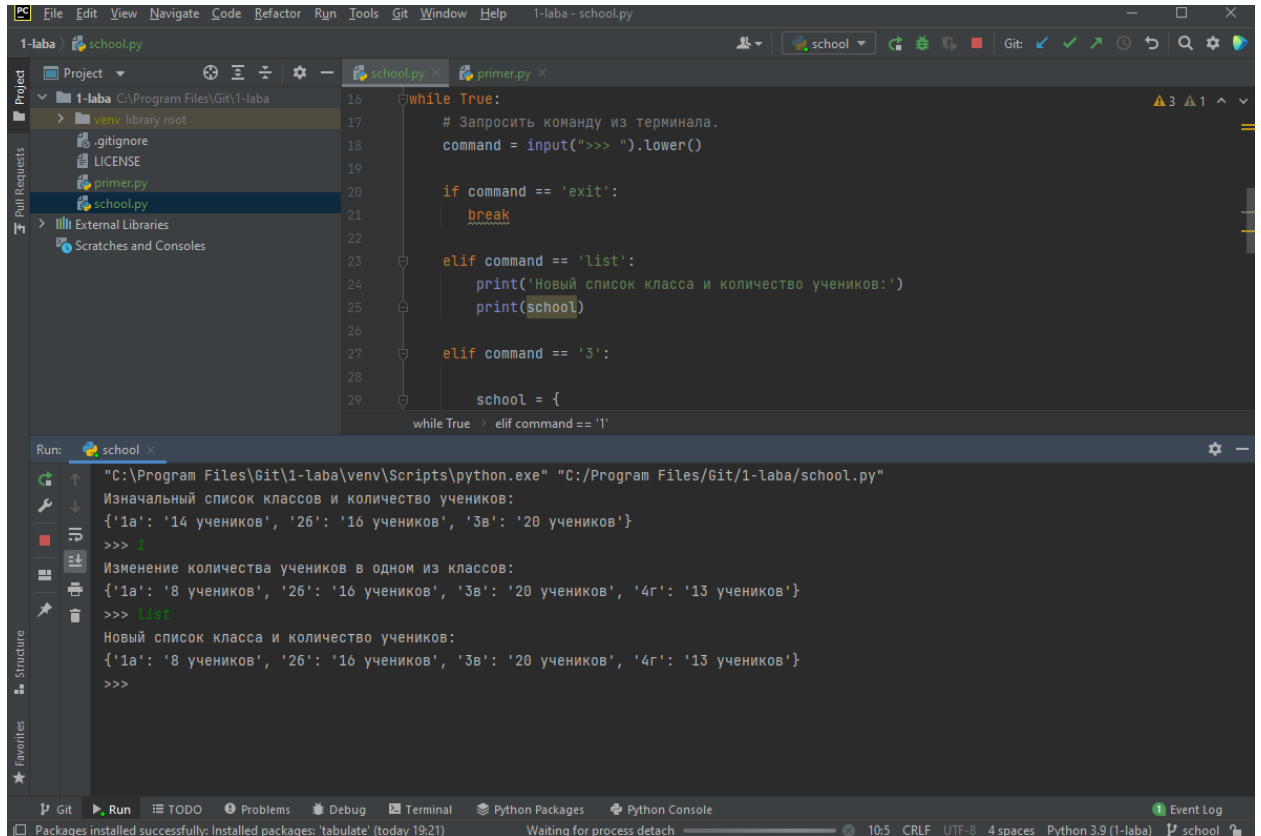
Run: primer

```
>>>
Фамилия и инициалы? Гончаров Д. П.
Должность?
Год поступления? 2020
>>>
Фамилия и инициалы? Гончаров Д. П.
Должность?
Год поступления? 2020
>>>
1: Гончаров Д. П.
>>>
```

Рисунок 3. Пример лабораторной работы

Задание 2. 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.



The screenshot shows an IDE with a Python script named `school.py` and its execution output in the Run console.

Code in `school.py`:

```
16 while True:
17     # Запросить команду из терминала.
18     command = input(">>> ").lower()
19
20     if command == 'exit':
21         break
22
23     elif command == 'list':
24         print('Новый список класса и количество учеников:')
25         print(school)
26
27     elif command == '3':
28         school = {
29             while True: > elif command == '1'
```

Run console output:

```
"C:\Program Files\Git\1-laba\venv\Scripts\python.exe" "C:/Program Files/Git/1-laba/school.py"
Изначальный список классов и количество учеников:
{'1а': '14 учеников', '2б': '16 учеников', '3в': '20 учеников'}
>>> 
Изменение количества учеников в одном из классов:
{'1а': '8 учеников', '2б': '16 учеников', '3в': '20 учеников', '4г': '13 учеников'}
>>> list
Новый список класса и количество учеников:
{'1а': '8 учеников', '2б': '16 учеников', '3в': '20 учеников', '4г': '13 учеников'}
>>>
```

Рисунок 4. Задание 2 (а)

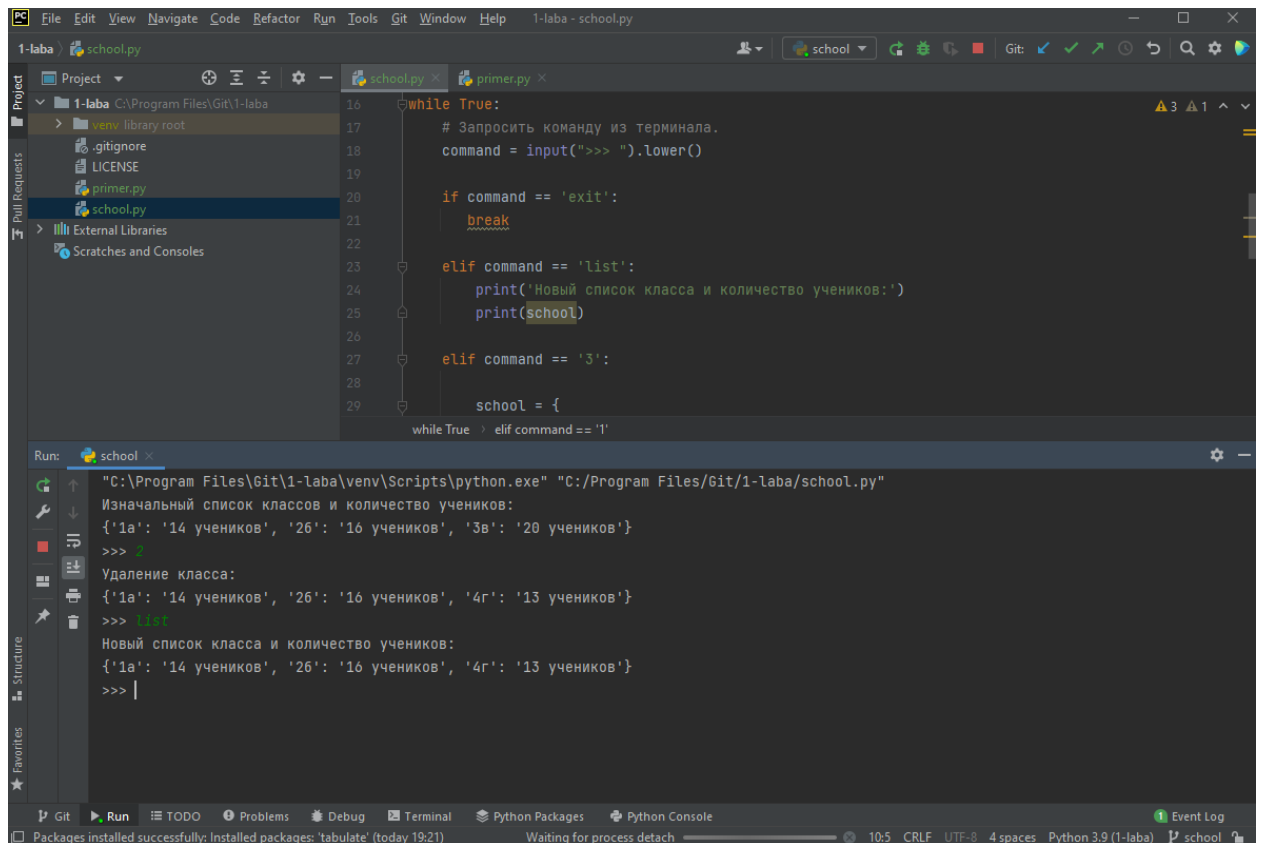


Рисунок 5. Задание 2 (б)

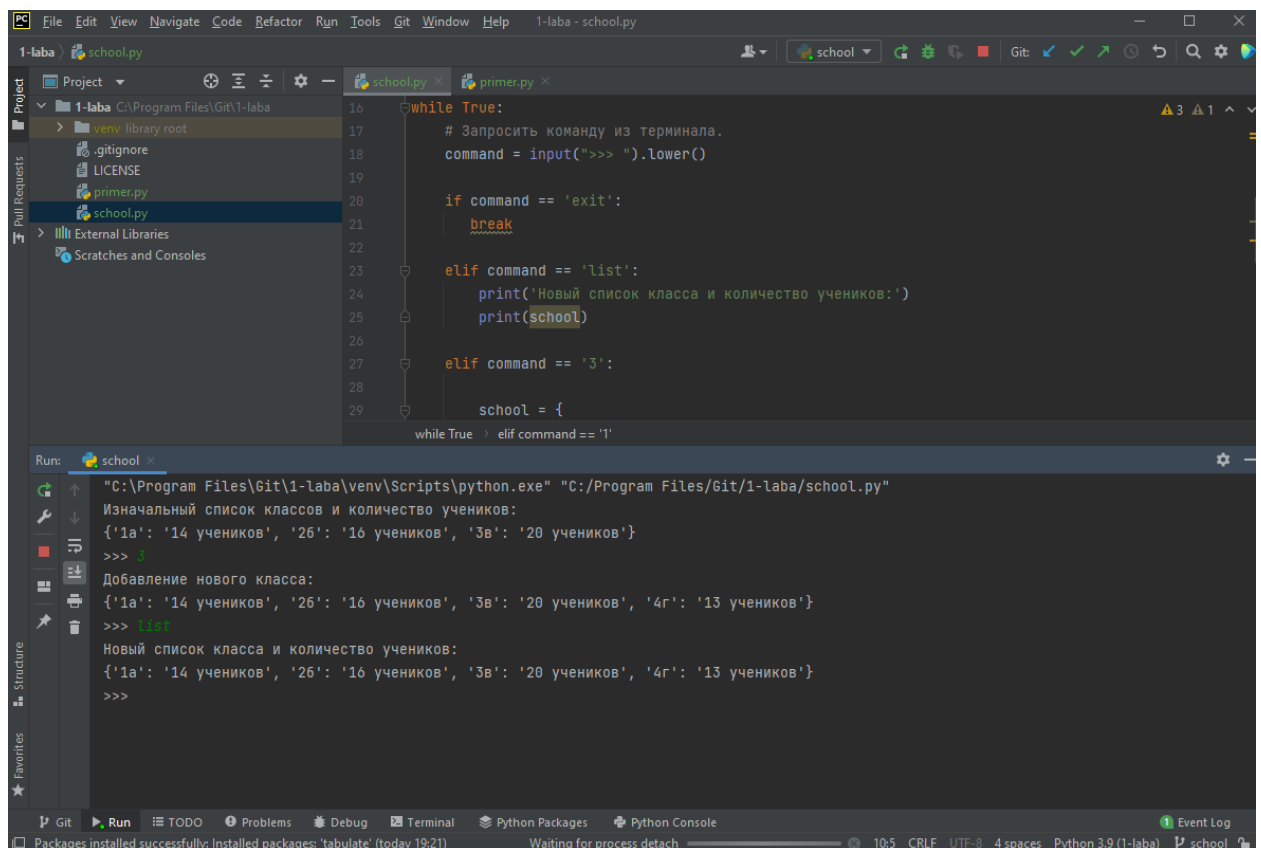
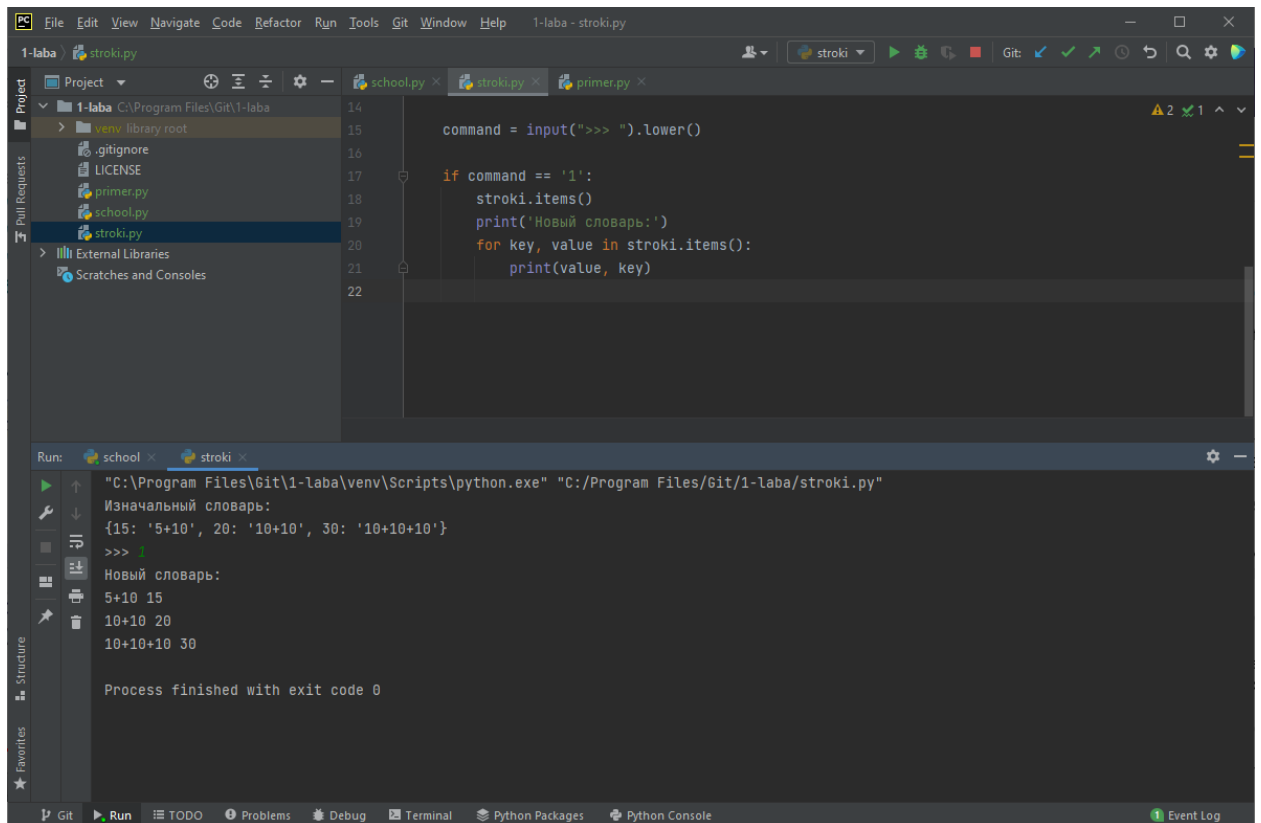


Рисунок 6. Задание 2 (в)

Задание 3. Решите задачу: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод `items()`, и с помощью полученного объекта `dict_items` создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.



The screenshot shows an IDE window with a Python file named `stroki.py`. The code in the editor is as follows:

```
14  
15  
16 command = input(">>> ").lower()  
17  
18 if command == '1':  
19     stroki.items()  
20     print('Новый словарь:')  
21     for key, value in stroki.items():  
22         print(value, key)
```

The Run console at the bottom shows the execution of the script. It starts with the command prompt `"C:\Program Files\Git\1-laba\venv\Scripts\python.exe" "C:/Program Files/Git/1-laba/stroki.py"`. The output shows the initial dictionary: `Изначальный словарь: {15: '5+10', 20: '10+10', 30: '10+10+10'}`. After the user enters `>>> 1`, it shows the new dictionary: `Новый словарь: 5+10 15, 10+10 20, 10+10+10 30`. The process finished with exit code 0.

Рисунок 7. Задание 3

Индивидуальное задание (Вариант 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.

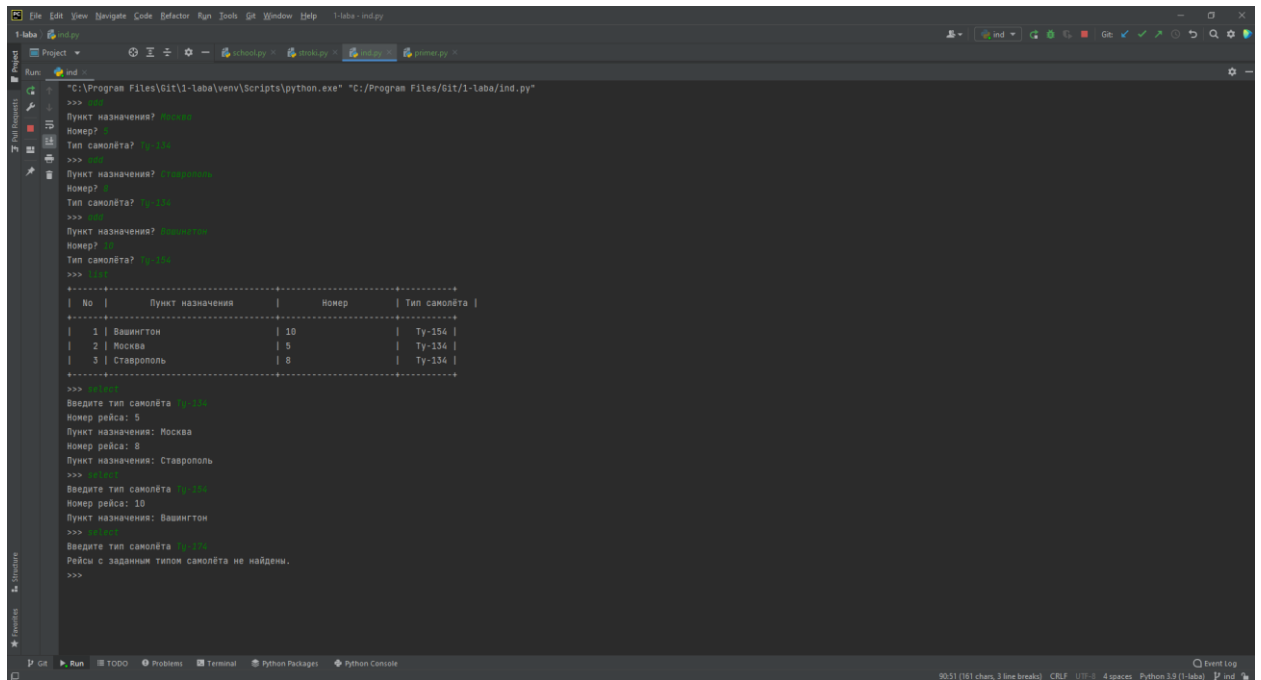


Рисунок 8. Индивидуальное задание

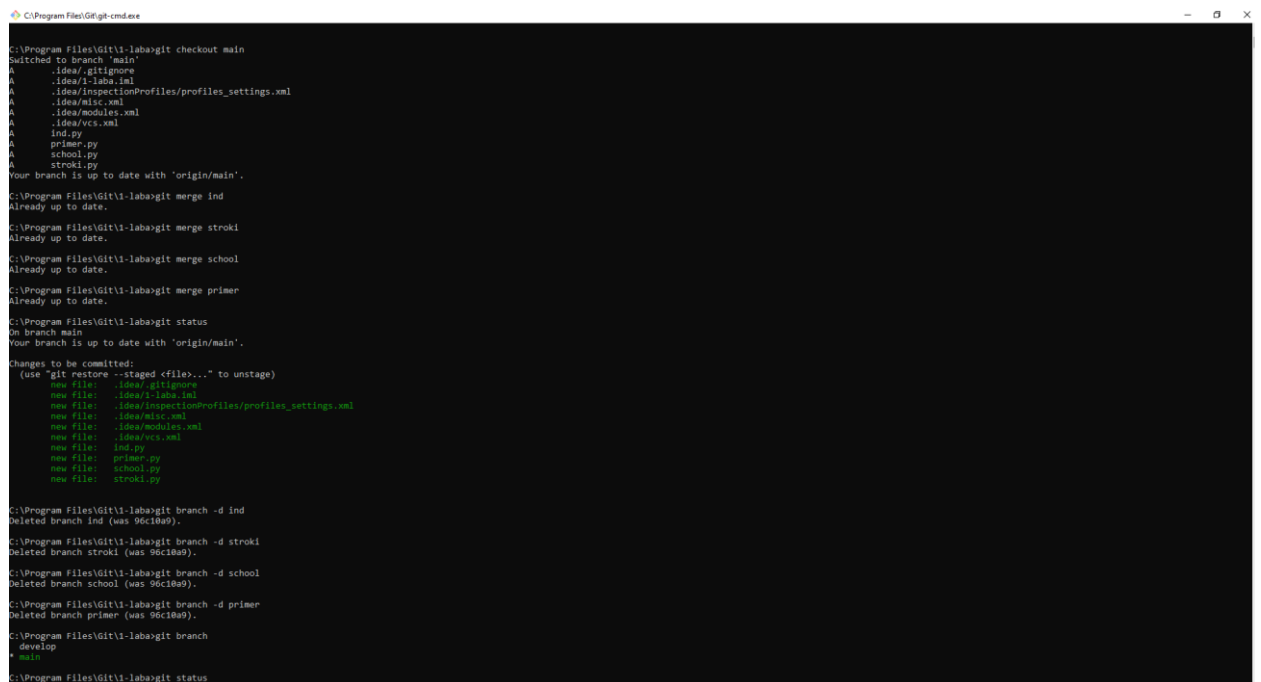


Рисунок 9. Слияние веток для разработок с веткой master

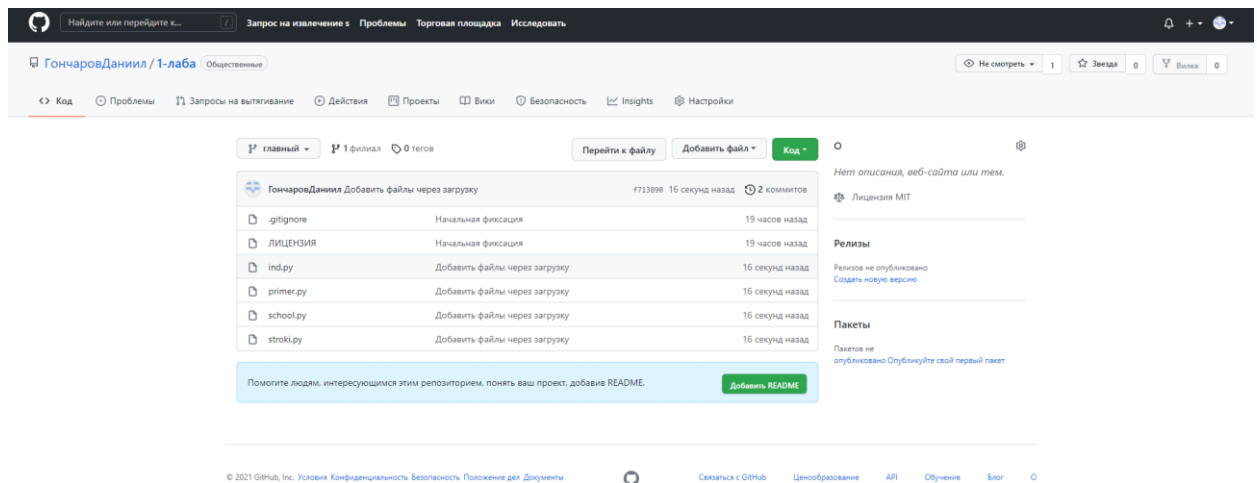


Рисунок 10. Отправление изменений на сервер GitHub

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