

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 №2.6 (9)
дисциплины
«Программирование на Python»

Выполнил:
Ибрагимов Муса Айнудинович
2 курс, группа ИВТ-6-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:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

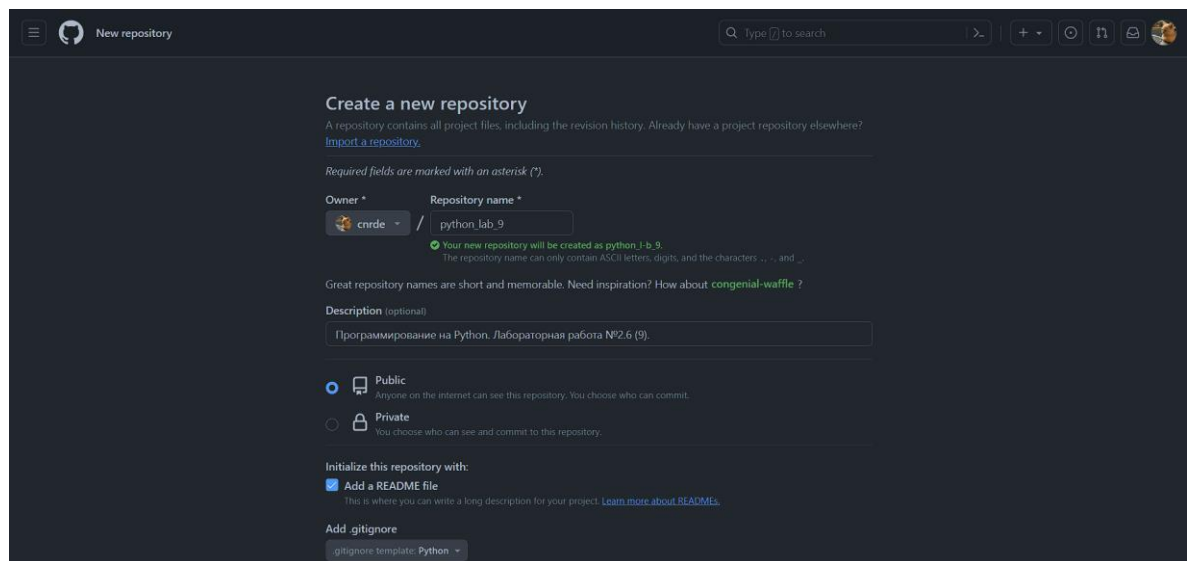
Ставрополь, 2024 г.

Тема: Работа со словарями в языке Python.

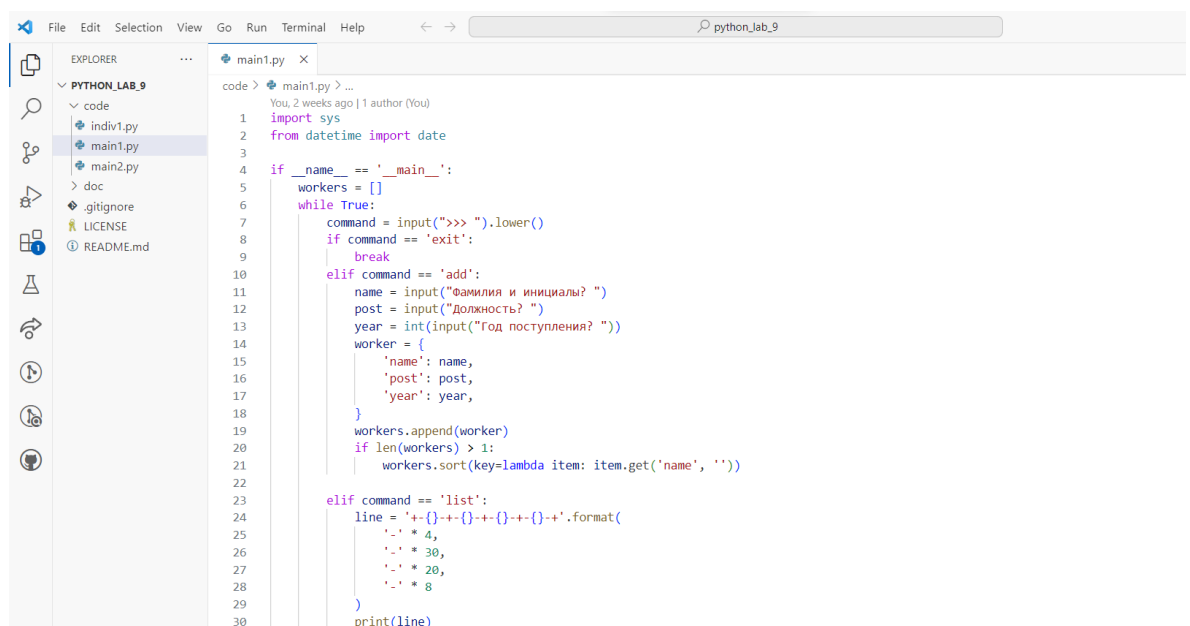
Цель: Приобретение навыков по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.

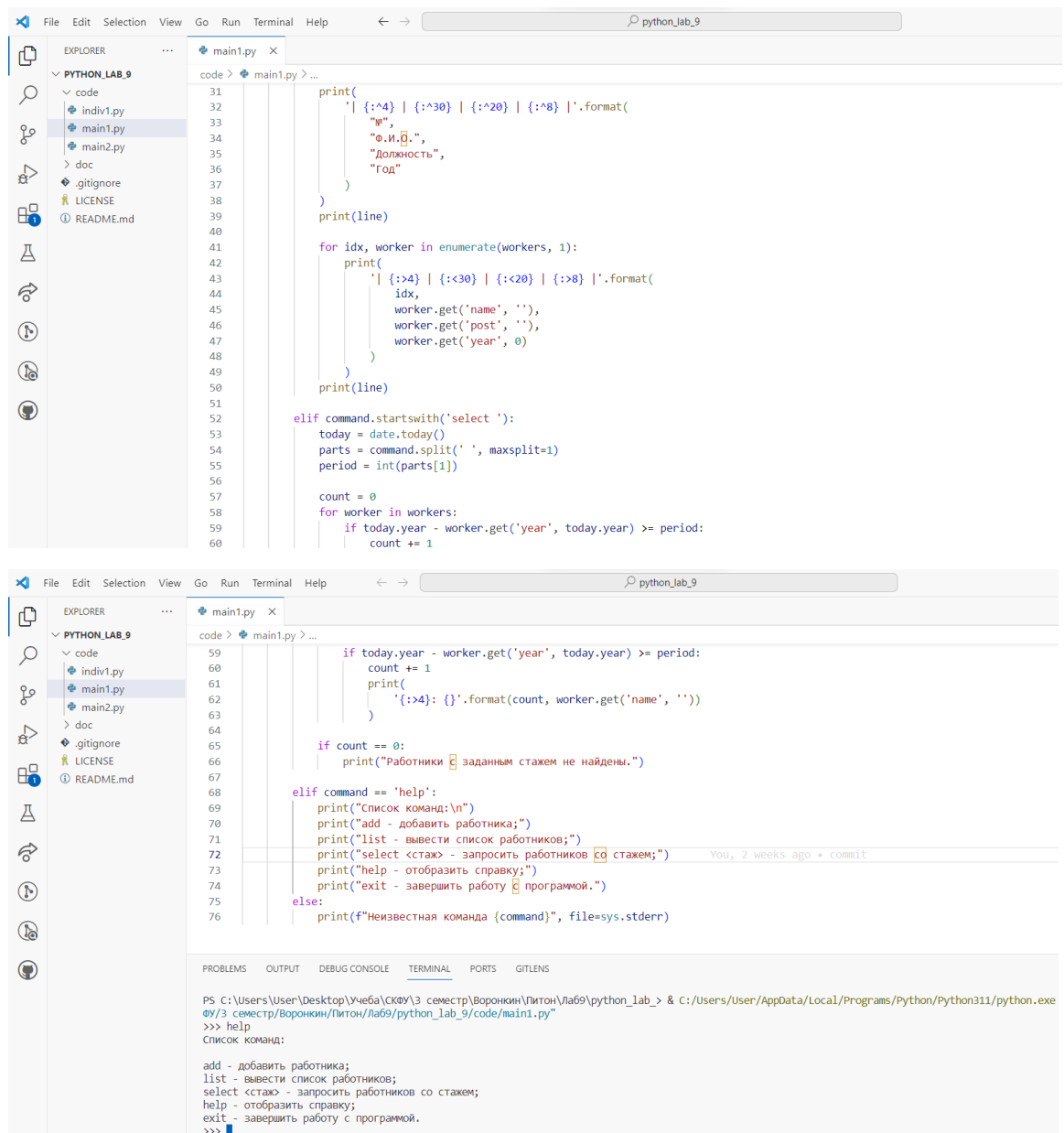
Практическая часть:

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



3. Проработал примеры.





4. Решил 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,
- в) в школе был расформирован (удален) другой класс.

Вычислите общее количество учащихся в школе.

The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'PYTHON_LAB_9'. The file explorer shows a 'code' folder containing 'main1.py' and 'main2.py'. The 'main2.py' file is selected and its code is visible in the editor. The code defines a dictionary 'school' with keys '1a', '2b', '11a', and '11b', and a function 'del school['2b']' which is then used to calculate the sum of the values. The terminal at the bottom shows the command 'python.exe' being executed, resulting in the output: '{ '1a': 59, '11a': 15, '11b': 10 }' and 'Количество учеников: 84'.

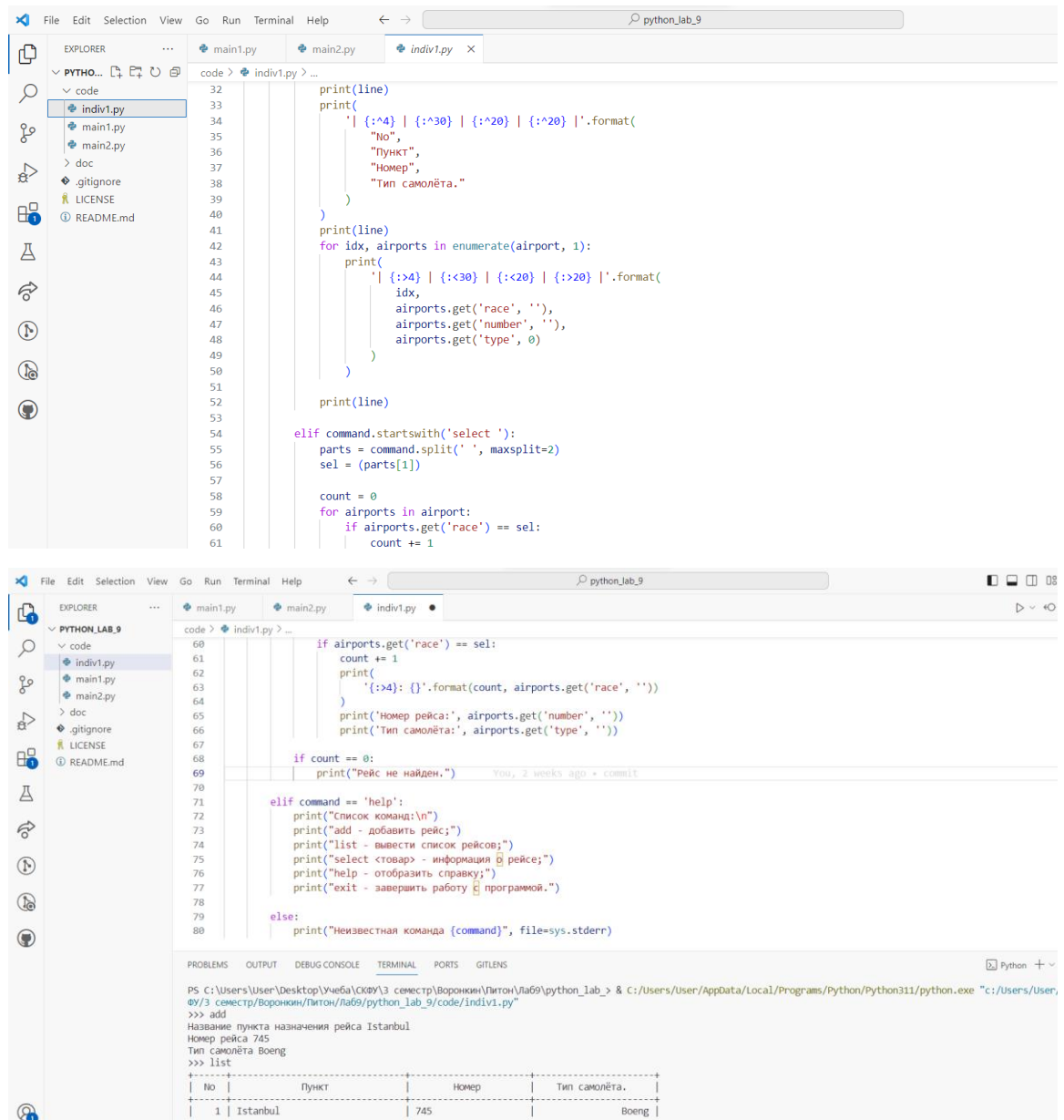
```
1 if __name__ == '__main__':
2     # 1
3     school = {'1a': 20, '2b': 24, '11a': 15, '11b': 10}
4     school['1a'] = 59
5     school['2b'] = 127
6
7     del school['2b']
8     s = sum(school.values())
9     print(school, "\nколичество учеников: ", s, '\n')
10
11 # 2
12 a = {1: 'one', 2: 'two', 3: 'three'}
13 inverse_a = {}
14 for k, v in a.items():
15     inverse_a[v] = k
16 print(a)
17 print(inverse_a)
```

PS C:\Users\User\Desktop\учеба\КФУ\3 семестр\Воронкин\Питон\лаб9\python_lab_9> & C:/Users/User/AppData/Local/Programs/Python/Python311/python.exe КФУ/3 семестр/Воронкин/Питон/лаб9/python_lab_9/code/main2.py
{ '1a': 59, '11a': 15, '11b': 10 }
Количество учеников: 84
{1: 'one', 2: 'two', 3: 'three'}
{ 'one': 1, 'two': 2, 'three': 3 }
PS C:\Users\User\Desktop\учеба\КФУ\3 семестр\Воронкин\Питон\лаб9\python_lab_9>

5. Выполнил индивидуальное задание.

The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying a project named 'PYTHON_LAB_9'. The file explorer shows a 'code' folder containing 'main1.py', 'main2.py', and 'indiv1.py'. The 'indiv1.py' file is selected and its code is visible in the editor. The code defines a dictionary 'airports' with keys 'race', 'number', and 'type', and a function 'airport.append(airports)' which is then used to calculate the sum of the values. The terminal at the bottom shows the command 'python.exe' being executed, resulting in the output: '{ '1a': 59, '11a': 15, '11b': 10 }' and 'Количество учеников: 84'.

```
1 import sys
2
3 if __name__ == '__main__':
4     airport = []
5     while True:
6         command = input(">>> ").lower()
7         if command == 'exit':
8             break
9
10    elif command == 'add':
11        race = input("Название пункта назначения рейса ")
12        number = input("Номер рейса ")
13        type = input("Тип самолёта ")
14
15        airports = {
16            'race': race,
17            'number': number,
18            'type': type,
19        }
20
21        airport.append(airports)
22        if len(airport) > 1:
23            airport.sort(key=lambda item: item.get('race', ''))
24
25    elif command == 'list':
26        line = '+{}-+{}-+{}-+{}-+'.format(
27            '-' * 4,
28            '-' * 30,
29            '-' * 20,
30            '-' * 20
31        )
```



Ответы на контрольные вопрос

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)
```

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

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