

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
дисциплины «Основы кроссплатформенного программирования»
Вариант ____

Выполнил:

Быковская Стефания Станиславовна
1 курс, группа ИТС-б-0-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», направ-
ленность (профиль) «Инфокоммуника-
ционные системы и сети», очная форма
обучения

(подпись)

Руководитель практики:

Воронкин Р.А. доцент кафедры инфо-
коммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

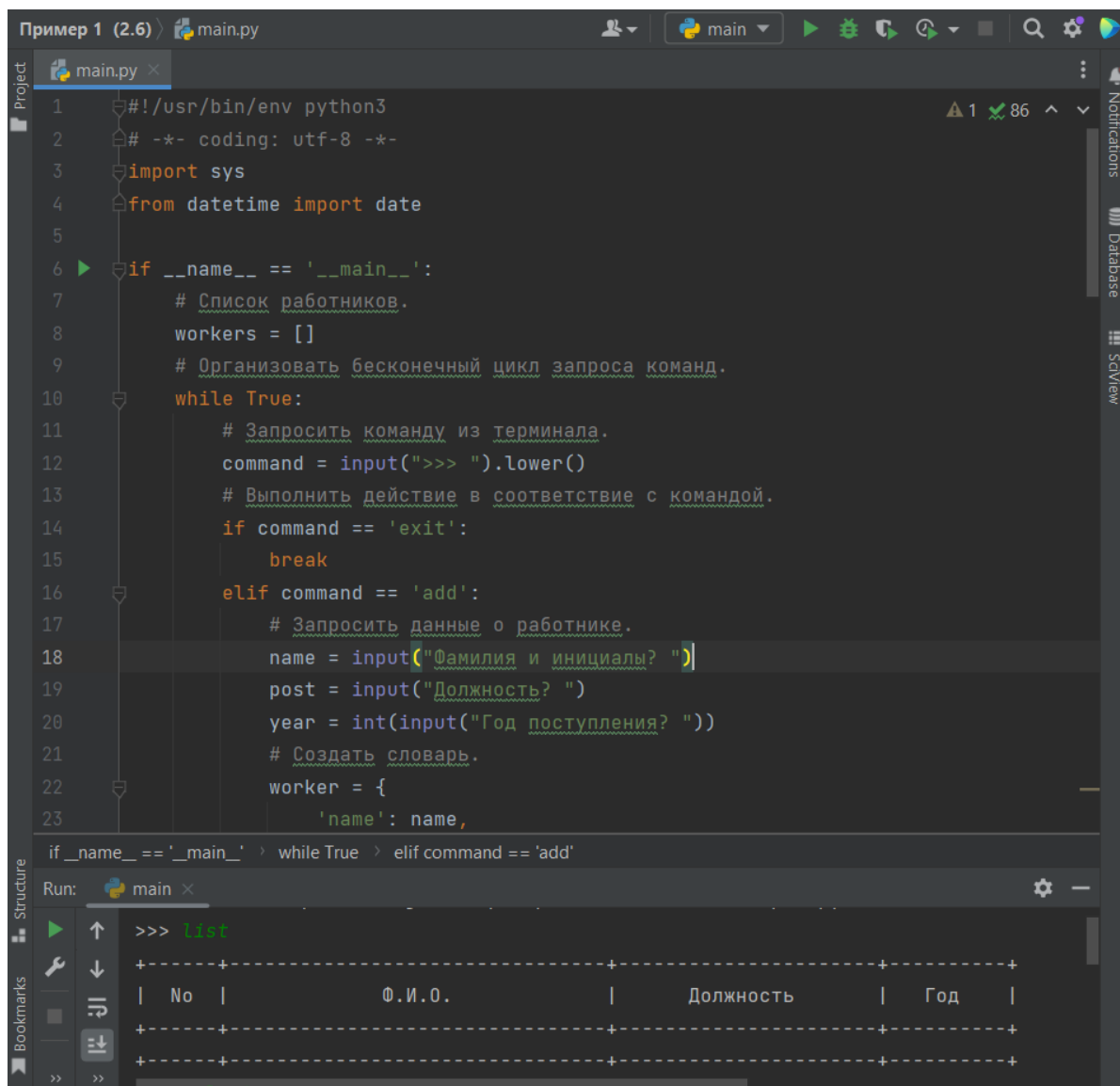
Ставрополь, 2023 г.

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.

Порядок выполнения работы:

Пример 1. Использовать словарь, содержащий следующие ключи: фамилия и инициалы работника; название занимаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия:



```
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      # Организовать бесконечный цикл запроса команд.
10     while True:
11         # Запросить команду из терминала.
12         command = input(">>> ").lower()
13         # Выполнить действие в соответствии с командой.
14         if command == 'exit':
15             break
16         elif command == 'add':
17             # Запросить данные о работнике.
18             name = input("Фамилия и инициалы? ")
19             post = input("Должность? ")
20             year = int(input("Год поступления? "))
21             # Создать словарь.
22             worker = {
23                 'name': name,
```

if __name__ == '__main__': > while True > elif command == 'add'

Run: main ×

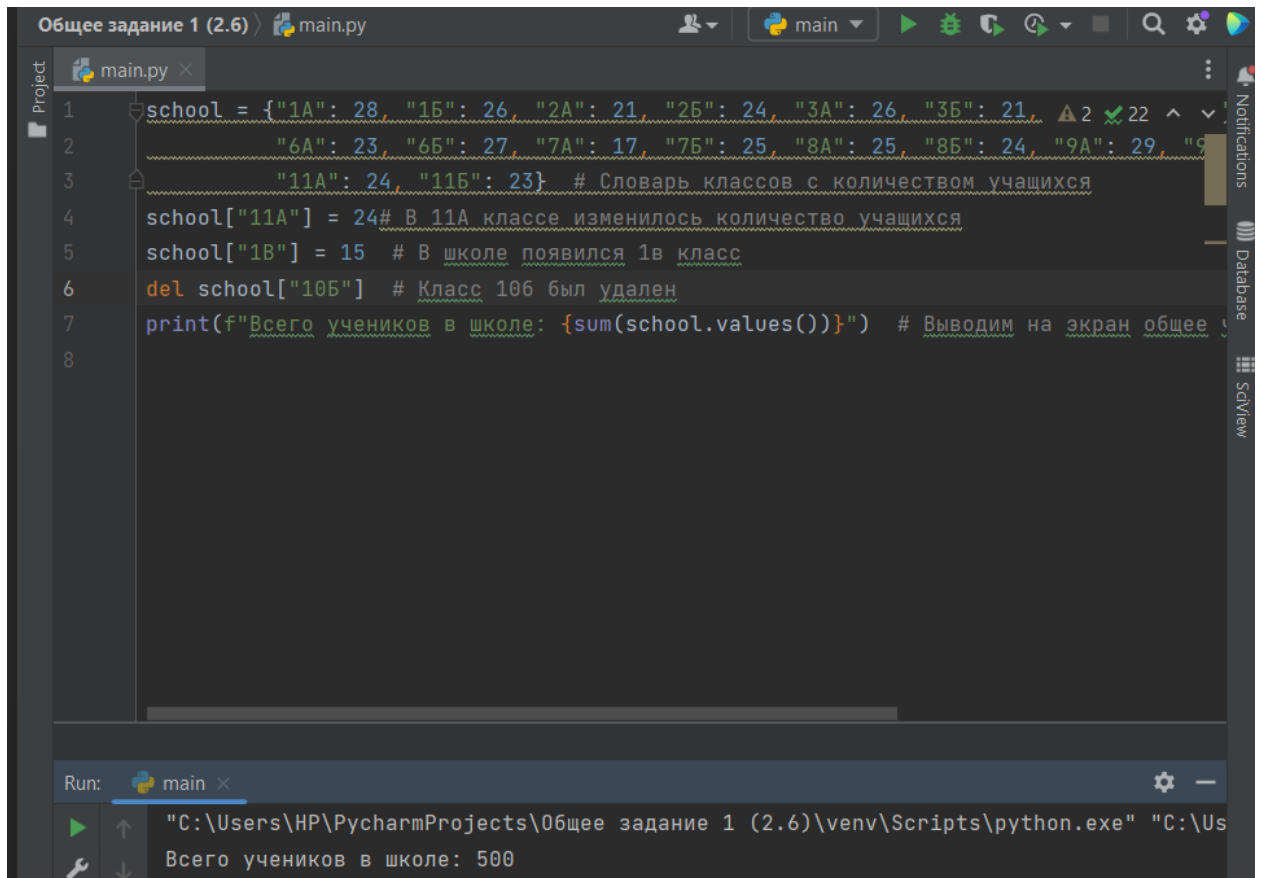
```
>>> List
+-----+-----+-----+-----+
| No |          Ф.И.О.          |      Должность      |      Год      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

Рисунок 1 – Выполнение примера №1.

Общее задание 1.

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.



```
1 school = {"1А": 28, "1Б": 26, "2А": 21, "2Б": 24, "3А": 26, "3Б": 21,
2          "6А": 23, "6Б": 27, "7А": 17, "7Б": 25, "8А": 25, "8Б": 24, "9А": 29, "9Б": 28,
3          "11А": 24, "11Б": 23} # Словарь классов с количеством учащихся
4 school["11А"] = 24 # В 11А классе изменилось количество учащихся
5 school["1Б"] = 15 # В школе появился 1в класс
6 del school["10Б"] # Класс 10Б был удален
7 print(f"Всего учеников в школе: {sum(school.values())}") # Выводим на экран общее
8
```

Run: main ×

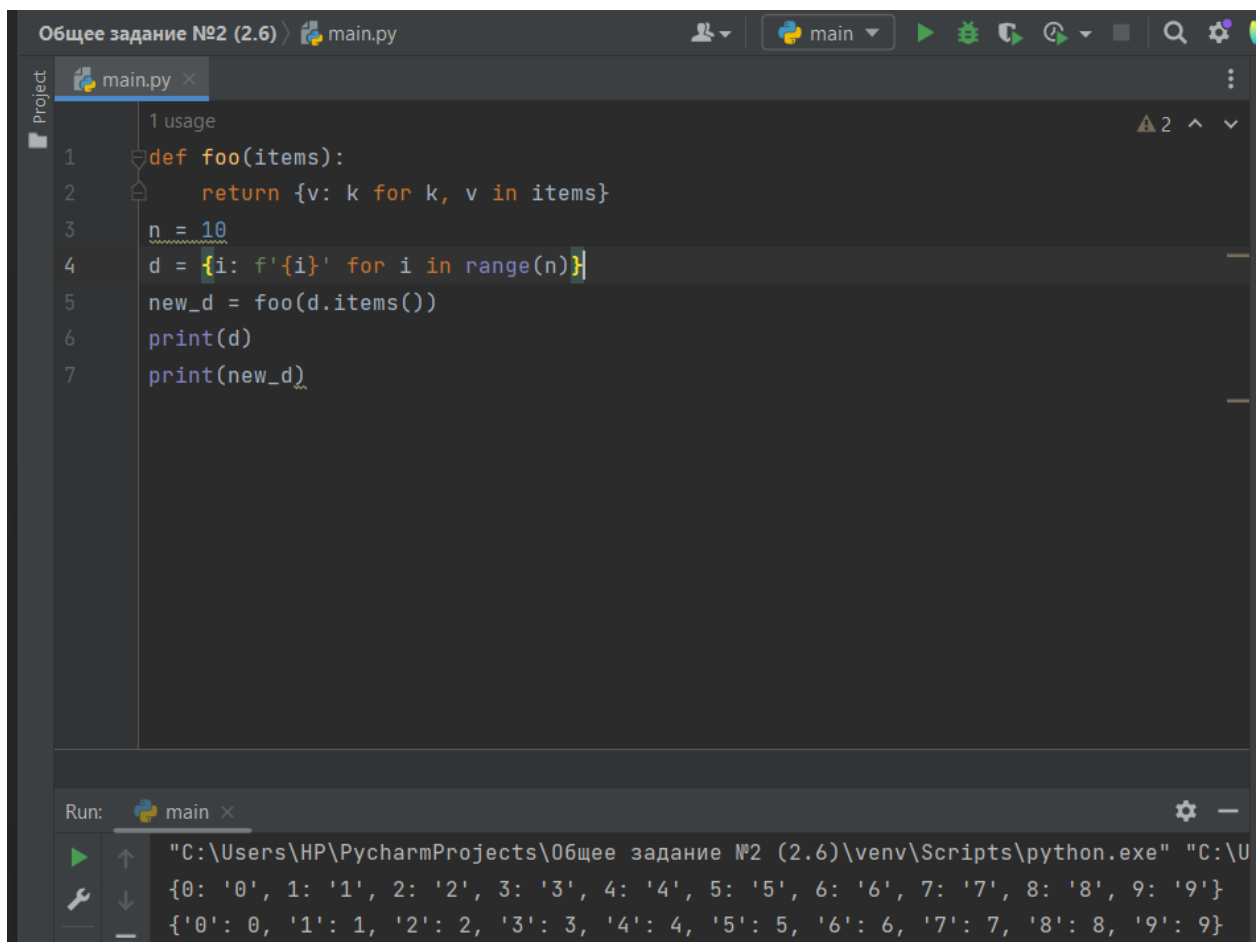
"C:\Users\HP\PycharmProjects\Общее задание 1 (2.6)\venv\Scripts\python.exe" "C:\Users\HP\PycharmProjects\Общее задание 1 (2.6)\main.py"

Всего учеников в школе: 500

Рисунок 2 – Выполнение общего задания №1.

Общее задание 2.

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 the PyCharm IDE interface. The top toolbar includes icons for running, debugging, and other development tools. The main editor window displays a Python file named `main.py` with the following code:

```
1 usage
2
3 def foo(items):
4     return {v: k for k, v in items}
5
6 n = 10
7 d = {i: f'{i}' for i in range(n)}
8
9 new_d = foo(d.items())
10
11 print(d)
12
13 print(new_d)
```

Below the editor, the 'Run' console shows the output of the program:

```
Run: main x
"C:\Users\HP\PycharmProjects\Общее задание №2 (2.6)\venv\Scripts\python.exe" "C:\U
{0: '0', 1: '1', 2: '2', 3: '3', 4: '4', 5: '5', 6: '6', 7: '7', 8: '8', 9: '9'}
{'0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5, '6': 6, '7': 7, '8': 8, '9': 9}
```

Рисунок 3 – Выполнение общего задания №2.

Индивидуальное задание

Вариант 5

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.

The screenshot shows a PyCharm IDE with a Python file named `main.py`. The code implements a flight management system with a `while True` loop. It handles commands like `list` (to display a table of flights) and `add` (to add a new flight). The table has columns for No, Пункт, Номер, and Тип самолёта. The Run window shows the execution of the script, with the `list` command being entered and the corresponding table being displayed.

```
1 import sys
2
3 if __name__ == '__main__':
4     # Список .
5     airport = []
6
7     # Организовать бесконечный цикл запроса команд.
8     while True:
9         # Запросить команду из терминала.
10        command = input(">>> ").lower()
11
12        # Выполнить действие в соответствие с командой.
13        if command == 'exit':
14            break
15
16        elif command == 'add':
17            # Запросить данные .
18            place = input("Название пункта назначения рейса ")
19            number = input("Номер рейса ")
20
21 if __name__ == '__main__': > while True > elif command == 'add'
```

Run: main ×

"C:\Users\HP\PycharmProjects\ ИД31 (2.6)\venv\Scripts\python.exe" "C:\Users\HP\PycharmProjects\ ИД31 (2.6)"

>>> list

No	Пункт	Номер	Тип самолёта.

>>> add

Название пункта назначения рейса

Номер рейса

Тип самолёта

Рисунок 4 – Выполнение ИДЗ.

Ответы на контрольные вопросы

1) Что такое словари в языке Python?

В языке Python словари - это изменяемый тип данных, который представляет собой неупорядоченную коллекцию пар ключ-значение. Ключи должны быть уникальными и неизменяемыми (например, строки или числа), а значения могут быть любого типа данных (строки, числа, списки, другие словари и т.д.).

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

Да, функция `len()` может быть использована при работе со словарями. Она возвращает количество элементов (пар ключ-значение) в словаре.

3) Какие методы обхода словарей Вам известны?

В Python есть несколько способов обхода словарей: с помощью цикла `for` можно перебрать все ключи или все значения словаря, а также можно использовать методы словаря, такие как `key()`, `values()` и `items()`, чтобы полу-

чить соответственно список ключей, список значений или список кортежей (ключ, значение).

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 создает 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 `zi()` 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.

Выводы: приобрела навыки по работе со словарями при написании программ с помощью языка программирования Python 3.x.