МИНИCTEPCTBO 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 «Теории программирования и алгоритмизации»**

Выполнил студент группы ИВТ-б-о-20-1

Плотников Д. В. « »\_\_\_\_\_\_\_\_\_20\_\_г.

Подпись студента\_\_\_\_\_\_\_\_\_\_\_

Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Проверил 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**

1. Создал новый собственный репозиторий. Ссылка на репозиторий: https://github.com/Dmitry-15/8\_laba.git.
2. С помощью команды git clone клонировал удаленный репозиторий на свой ПК. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.
3. Проработал пример лабораторной работы.

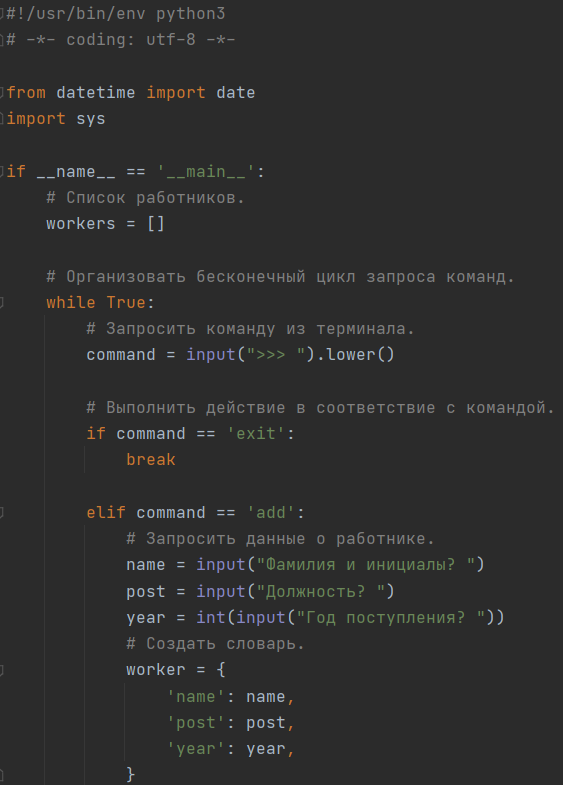


Рисунок 1. Код примера

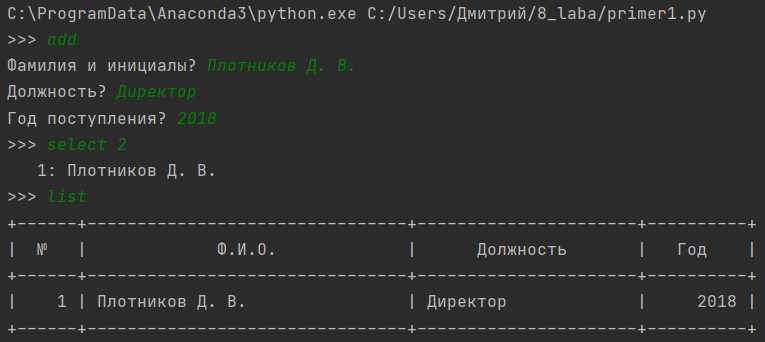


Рисунок 2. Выполнение примера

**Задание 1**

1. Условие задачи: 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.

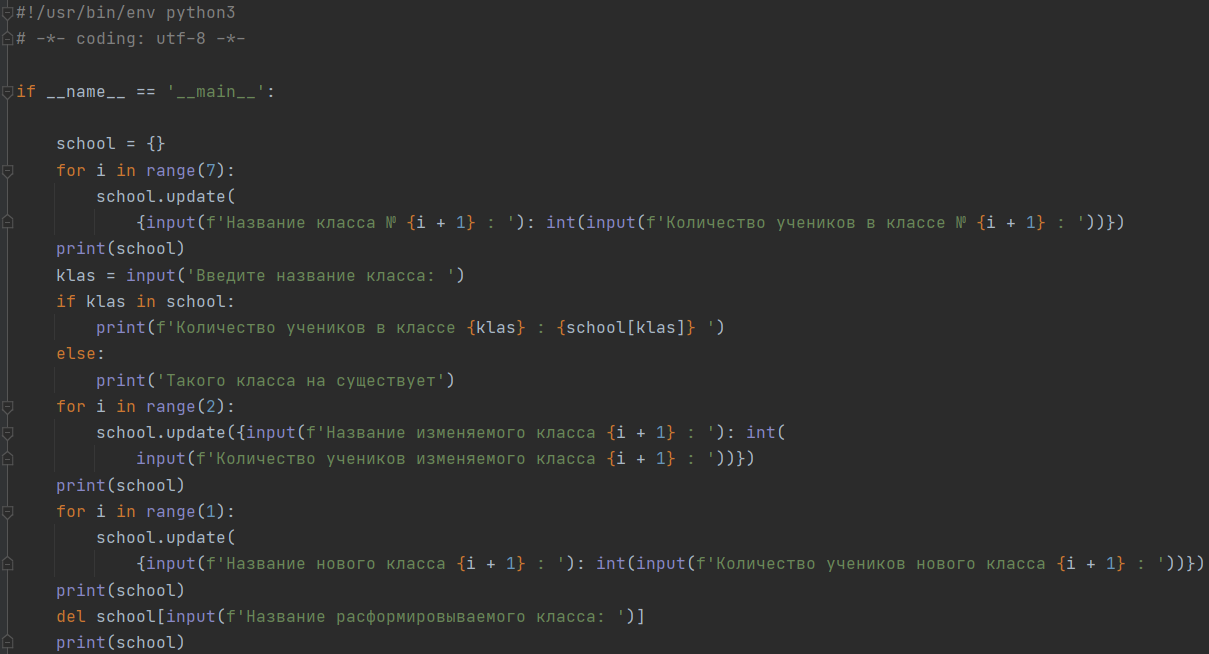


Рисунок 3. Код задания 1

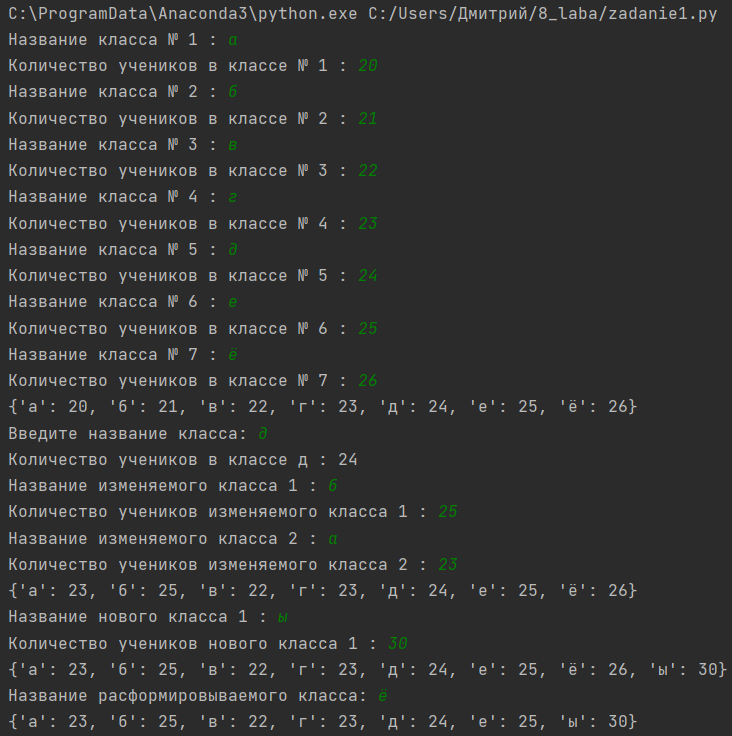


Рисунок 4. Выполнение задания 1

**Задание 2**

1. Условие задачи: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод *items()*, c с помощью полученного объекта dict\_items создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.

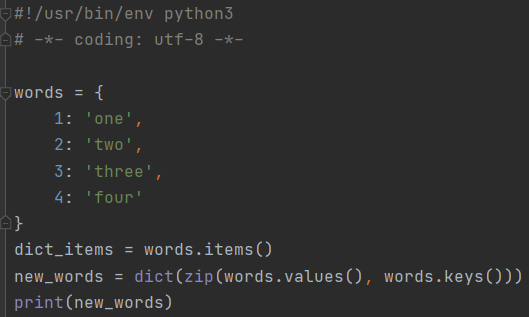


Рисунок 5. Код задания 2

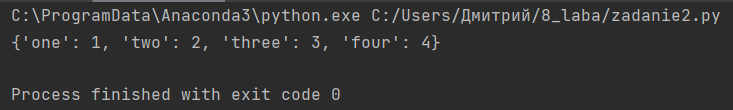


Рисунок 6. Выполнение задания 2

**Индивидуальное задание**

**Вариант 14**

1. Условие задачи: Использовать словарь, содержащий следующие ключи: фамилия, имя; знак Зодиака; дата рождения (список из трех чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по датам рождения; вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.



Рисунок 7. Код индивид. Задания

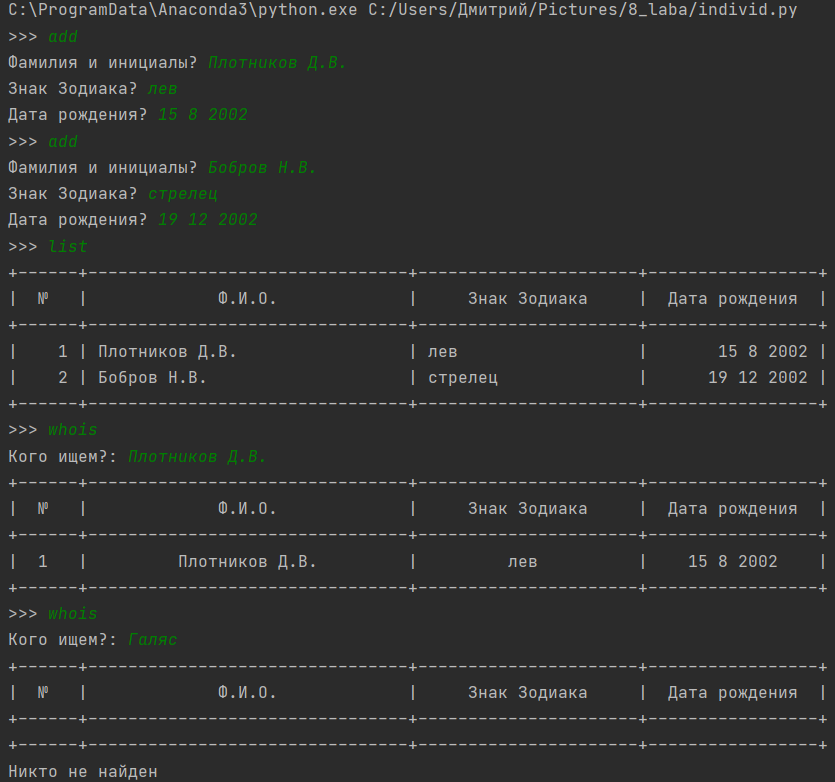


Рисунок 8. Выполнение индивид. задания

**Контрольные вопросы**

1. Что такое словари в языке Python?

Словарь – это структура данных, которая предназначена для хранения произвольных объектов с доступом по ключу.

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

Да, может.

1. Какие методы обхода словарей Вам известны?

* For I in dict
* for key in dict.keys()
* for value in dict.value()
* for key, value in dict.items()

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

Print(dict[‘Название ключа’])

Print(dict.get(‘Название ключа’))

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

Dict[‘Ключ’]= значение

1. Что такое словарь исключений?

Исключение в Python – это конструкция, используемая для сигнализации о важном событии, обычно об ошибке, которое происходит при выполнении программы. Исключение может привести к остановке программы, если она не будет должным образом «поймана» (т.е. обработана правильно). Если вы думаете, что ваша программа может вызвать исключение при выполнении.

1. Самостоятельно изучите возможности функции zip() приведите примеры ее использования.

Функция zip объединяет в кортежи элементы из последовательностей переданных в качестве аргументов. Функция прекращает выполнение, как только достигнут конец самого короткого списка.

a = [1,2,3]

b = "xyz"

c = (None, True)

res = list(zip(a, b, c))

print (res)

[(1, 'x', None), (2, 'y', True)]

1. 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.

Класс datetime.date(year, month, day) - стандартная дата. Атрибуты: year, month, day. Неизменяемый объект.

Класс datetime.time(hour=0, minute=0, second=0, microsecond=0, tzinfo=None) - стандартное время, не зависит от даты. Атрибуты: hour, minute, second, microsecond, tzinfo.

Класс datetime.timedelta - разница между двумя моментами времени, с точностью до микросекунд.

Класс datetime.tzinfo - абстрактный базовый класс для информации о временной зоне (например, для учета часового пояса и / или летнего времени).

Класс datetime.datetime(year, month, day, hour=0, minute=0, second=0, microsecond=0, tzinfo=None) - комбинация даты и времени.

Обязательные аргументы:

* datetime.MINYEAR (1) ≤ year ≤ datetime.MAXYEAR (9999)
* 1 ≤ month ≤ 12
* 1 ≤ day ≤ количество дней в данном месяце и году

Методы класса datetime:

* datetime.today() - объект datetime из текущей даты и времени. Работает также, как и datetime.now() со значением tz=None.
* datetime.fromtimestamp(timestamp) - дата из стандартного представления времени.
* datetime.fromordinal(ordinal) - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970.
* datetime.now(tz=None) - объект datetime из текущей даты и времени.
* datetime.combine(date, time) - объект datetime из комбинации объектов date и time.
* datetime.strptime(date\_string, format) - преобразует строку в datetime (так же, как и функция strptime из модуля time).
* datetime.strftime(format) - см. функцию strftime из модуля time.
* datetime.date() - объект даты (с отсечением времени).
* datetime.time() - объект времени (с отсечением даты).
* datetime.replace([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]]]) - возвращает новый объект datetime с изменёнными атрибутами.
* datetime.timetuple() - возвращает struct\_time из datetime.
* datetime.toordinal() - количество дней, прошедших с 01.01.1970.
* datetime.timestamp() - возвращает время в секундах с начала эпохи.
* datetime.weekday() - день недели в виде числа, понедельник - 0, воскресенье - 6.
* datetime.isoweekday() - день недели в виде числа, понедельник - 1, воскресенье - 7.
* datetime.isocalendar() - кортеж (год в формате ISO, ISO номер недели, ISO день недели).
* datetime.isoformat(sep='T') - красивая строка вида "YYYY-MM-DDTHH:MM:SS.mmmmmm" или, если microsecond == 0, "YYYY-MM-DDTHH:MM:SS"
* datetime.ctime() - см. ctime() из модуля time.

**Вывод:** в ходе выполнения лабораторной работы успешно приобрел навыки по работе со словарями при написании программ с помощью языка программирования Python3.