

Северо-Кавказский федеральный университет
Институт математики и информационных технологий

ОТЧЕТ
о выполнении лабораторной работы №4
по дисциплине
«Основы Программной Инженерии»

Выполнил:

Маняхин Тимур Александрович

студент 2 курса, ПИЖ-б-о-20-1 группы бакалавриата
«Программная инженерия»

очной формы обучения

Ставрополь, 2021

Северо-Кавказский федеральный университет
Институт математики и информационных технологий

Скриншоты работы программ

```
if __name__ == '__main__':
    school = {"1a": 23, "1б": 22, "2a": 21, "2б": 20, "3a": 19, "3б": 20, "4a": 18, "4б": 24, "5a": 23, "5б": 17,
              "6a": 20, "6б": 15, "7a": 18, "7б": 25, "8a": 25, "8б": 26, "9a": 29, "9б": 17, "10a": 24, "10б": 22,
              "11a": 24, "11б": 23}

    school["2a"] = 19

    school["1б"] = 15

    del school["10б"]

    print(f"Всего учеников в школе: {sum(school.values())}")
```

Рисунок 1.1 – программа 1

Всего учеников в школе: 466

Рисунок 1.2 – результат работы программы 1

```
if __name__ == '__main__':
    def rev_key(dct):
        dct_new = dict()
        for i, v in dct.items():
            for w in v:
                dct_new[w] = dct_new.get(w, []) + [i]
        return dct_new

    dct = {1: 'acc', 2: 'cab', 3: 'ccb'}
    print(rev_key(dct))
```

Рисунок 1.1 – программа 2

{'a': [1, 2], 'c': [1, 1, 2, 3, 3], 'b': [2, 3]}

Рисунок 1.2 – результат работы программы 2

```
Введите номер рейса 122
Введите тип самолета airbus
>>> add
Введите название пункта назначения spb
Введите номер рейса 143
Введите тип самолета iboeing
>>> list
```

No	Пункт назначения	Номер рейса	Тип самолета
1	msk	122	airbus
2	spb	143	boeing

Рисунок 1.3 – результат работы программы iz

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Словарь представляет собой структуру данных (которая ещё называется ассоциативный массив), предназначенную для хранения произвольных объектов с доступом по ключу.
2. Да, `len()` может быть использован – он выводит количество элементов (пар типа «ключ: элемент»).
3. Перебор ключей в цикле `for`, перебор элементов в цикле `for`, одновременный перебор ключей и их значений в цикле `for`.
4. С помощью метода `get()`, при обходе в цикле `for`, используя переменную в качестве счетчика ключей.
5. С помощью метода `setdefault()`, при непосредственном обращении к ключу словаря.
6. Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка.
7. 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. Функция `zip()` принимает итерируемый объект, например, список, кортеж, множество или словарь в качестве аргумента. Затем она генерирует список кортежей, которые содержат элементы из каждого объекта, переданного в функцию. Предположим, что есть список имен и номером сотрудников, и их нужно объединить в массив кортежей. Для этого можно использовать функцию `zip()`.
8. 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. Классы, предоставляемые модулем `datetime`:
 1. Класс `datetime.date(year, month, day)` - стандартная дата. Атрибуты: `year`, `month`, `day`. Неизменяемый объект.
 2. 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`.
 3. Класс `datetime.timedelta` - разница между двумя моментами времени, с точностью до микросекунд.
 4. Класс `datetime.tzinfo` - абстрактный базовый класс для информации о временной зоне (например, для учета часового пояса и / или летнего времени).
 5. 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`
Необязательные: • `0 ≤ minute < 60` • `0 ≤ second < 60` • `0 ≤ microsecond < 1000000`Методы класса `datetime`:
 1. `datetime.today()` - объект `datetime` из текущей даты и времени. Работает также, как и `datetime.now()` со значением `tz=None`.
 2. `datetime.fromtimestamp(timestamp)` - дата из стандартного представления времени.
 3. `datetime.fromordinal(ordinal)` - дата из числа, представляющего собой количество дней, прошедших с 01.01.1970.
 4. `datetime.now(tz=None)` - объект `datetime` из текущей даты и времени.
 5. `datetime.combine(date, time)` - объект `datetime` из комбинации объектов `date` и `time`.
 6. `datetime.strptime(date_string, format)` - преобразует строку в `datetime` (так же, как и функция `strptime` из модуля `time`).
 7. `datetime.strftime(format)` - см. функцию `strftime` из модуля `time`.
 8. `datetime.date()` - объект даты (с отсечением времени).
 9. `datetime.time()` - объект времени (с отсечением даты).
 10. `datetime.replace([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]])` - возвращает новый объект `datetime` с изменёнными атрибутами.
 11. `datetime.timetuple()` - возвращает `struct_time` из `datetime`.
 12. `datetime.toordinal()` -

количество дней, прошедших с 01.01.1970. 13.datetime.timestamp() - возвращает время в секундах с начала эпохи. 14.datetime.weekday() - день недели в виде числа, понедельник - 0, воскресенье - 6. 15.datetime.isoweekday() - день недели в виде числа, понедельник - 1, воскресенье - 7. 16.datetime.isocalendar() - кортеж (год в формате ISO, ISO номер недели, ISO день недели). 17.datetime.isoformat(sep='T') - красивая строка вида "YYYY-MMDDTHH:MM:SS.mmmmmm" или, если microsecond == 0, "YYYY-MM-DDTHH:MM:SS" 18.datetime.ctime() - преобразует время, выраженное в секундах с начала эпохи в строку вида "Thu Sep 27 16:42:37 2012".