

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Работа со словарями в языке Python»

ОТЧЕТ
по лабораторной работе №9
дисциплины
«Основы программной инженерия»

Выполнил:

Зиёдуллаев Жавохир Эркин угли
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
from datetime import date

if __name__ == '__main__':
    # Список работников.
    workers = []

    # Организовать бесконечный цикл запроса команд.
    while True:
        # Запросить команду из терминала.
        command = input(">>> ").lower()

        # Выполнить действие в соответствие с командой.
        if command == 'exit':
            break

        elif command == 'add':
            # Запросить данные о работнике.
            name = input("Фамилия и инициалы? ")
            post = input("Должность? ")
```

```

post = input("Должность? ")
year = int(input("Год поступления? "))

# Создать словарь.
worker = {
    'name': name,
    'post': post,
    'year': year,
}

# Добавить словарь в список.
workers.append(worker)
# отсортировать список в случае необходимости.
if len(workers) > 1:
    workers.sort(key=lambda item: item.get('name', ''))

elif command == 'list':
    # Заголовок таблицы.
    line = '+--{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 8
    )

```

```

)
print(line)
print(
    '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
        "№",
        "Ф.И.О.",
        "Должность",
        "Год"
    )
)
print(line)

# Вывести данные о всех сотрудниках.
for idx, worker in enumerate(workers, 1):
    print(
        '| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
            idx,
            worker.get('name', ''),
            worker.get('post', ''),
            worker.get('year', 0)
        )
    )

```

```

    )
    print(line)
elif command.startswith('select '):
    # Получить текущую дату.
    today = date.today()

    # Разбить команду на части для выделения номера года.
    parts = command.split(' ', maxsplit=1)
    # Получить требуемый стаж.
    period = int(parts[1])

    # Инициализировать счетчик.
    count = 0
    # Проверить сведения работников из списка.
    for worker in workers:
        if today.year - worker.get('year', today.year) >= period:
            count += 1
            print(
                '{:>4}: {}'.format(count, worker.get('name', ''))
            )

    # Если счетчик равен 0, то работники не найдены.
    if count == 0:

```

```

    )

    # Если счетчик равен 0, то работники не найдены.
    if count == 0:
        print("Работники с заданным стажем не найдены. ")

elif command == 'help':
    # Вывести справку о работе с программой.
    print("Список команд:\n")
    print("add - добавить работника;")
    print("list - вывести список работников;")
    print("select <стаж> - запросить работников со стажем;")
    print("help - отобразить справку;")
    print("exit - завершить работу с программой.")

else:
    print(f"Неизвестная команда {command}", file=sys.stderr)

```

```

>>> add
Фамилия и инициалы? Турклиев В
Должность? Посудомойщик
Год поступления? 2022
>>> дшые
>>> Неизвестная команда дшые

>>> Неизвестная команда
list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Зиёдуллаев Ж | продавец | 2023 |
| 2 | Тарасов С | Стораж | 2021 |
| 3 | Турклиев В | Посудомойщик | 2022 |
+-----+-----+-----+-----+
>>> select 24
Работники с заданным стажем не найдены.
>>> select 1
1: Тарасов С
2: Турклиев В
>>> select 0
1: Зиёдуллаев Ж
2: Тарасов С
3: Турклиев В

```

Индивидуальное задание

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    # Список .
    airport = []

    # Организовать бесконечный цикл запроса команд.
    while True:
        # запросить команду из терминала.
        command = input(">>> ").lower()

        #Выполнить действие в соответствии с командой.
        if command == 'exit':
            break

        elif command == 'add':
            #Запросить данные.
            race = input("Название пункта назначения рейса ")
            number = input("Номер рейса ")

```

```

type = input("Тип самолёта ")
# создать словарь.
airport = {
    'race': race,
    'number': number,
    'type': type,
}

# Добавить словарь в список.
airport.append(airport)
# Отсортировать список в случае необходимости.
if len(airport) > 1:
    airport.sort(key=lambda item: item.get('race', ''))

elif command == 'list':
    # Заголовок таблицы.
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 20
    )

```

```

)
print(line)
print(
    '| {:^4} | {:^30} | {:^20} | {:^20} |'.format(
        "No",
        "Пункт",
        "Номер",
        "Тип самолёта."
    )
)
print(line)

# Вывести данные о всех рейсах.
for idx, airports in enumerate(airport, 1):
    print(
        '| {:>4} | {:<30} | {:<20} | {:>20} |'.format(
            idx,
            airports.get('race', ''),
            airports.get('number', ''),
            airports.get('type', 0)
        )
    )
)

```


Ссылка: <https://github.com/javoxir21/laba-9.git>

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Что такое словари в языке Python?

Словарь (dict) представляет собой структуру данных (которая ещё называется ассоциативный массив), предназначенную для хранения произвольных объектов с доступом по ключу. Данные в словаре хранятся в формате ключ – значение.

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

Да

- 3) Какие методы обхода словарей Вам известны?

for i in nums: print(nums[i]) for key, value in nums.items(): print(key, 'is', value)

Методы словаря keys() и values() позволяют получить отдельно перечни ключей и значений.

- 4) Какими способами можно получить значения из словаря по ключу?

dict["key"] dict.get("key")

- 5) Какими способами можно установить значение в словаре по ключу?

dict.setdefault("key", "value") dict["key"] = "value"

- 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. У функции zip() множество сценариев применения.

- 8) Самостоятельно изучите возможности модуля datetime. Каким функционалом по работе с датой и временем обладает этот модуль?

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.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.isocalendar() - кортеж (год в формате ISO, ISO номер недели, ISO день недели).

datetime.isoformat(sep='T') - красивая строка вида "YYYYMMDDTHH:MM:SS.mmmmmmm" или, если microsecond == 0, "YYYYMMDDTHH:MM:SS"