МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №6

Работа со словарями в языке Python.

по дисциплине «Технологии программирования и алгоритмизации»

Выполнил студент группы ИВТ-б-о-21-1

Лысенко И.А. « »_____20__г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2022

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

Ход работы:

1. Был создан следующий репозиторий: https://github.com/IsSveshuD/lab_8

2. Был проработан пример.

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

import ...

if __name__ == '__main__':
    workers = []

    while True:
        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? ")
            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:
            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)
```

Рисунок 1 – Пример

3. Было выполнено следующее индивидуальное задание.

Использовать словарь, содержащий следующие ключи:

– фамилия,

– имя;

– номер телефона;

– 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 трем первым цифрам номера телефона;

– вывод на экран информации о человеке, чья фамилия введена с клавиатуры;

– если такого нет, выдать на дисплей соответствующее сообщение.

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


import sys

if __name__ == '__main__':
    contacts = []

    while True:
        command = input(">>> ").lower()

        if command == 'exit':
            break

        elif command == 'add':

            family = input("Фамилия? ")
            name = input("Имя? ")
            number = int(input("Номер телефона? "))
            born = list(map(int, input("Дата рождения? ").split('.', 2)))

            contact = {
                'family': family,
                'name': name,
                'number': number,
                'born': born,
            }

            contacts.append(contact)
```

```
        if len(contacts) > 1:
            contacts.sort(key=lambda item: item.get('number', [0 - 2]))

    elif command == 'list':
        line = '+-{}-+-{}-+-{}-+-{}-+-{}-+'.format(
            '-' * 4,
            '-' * 30,
            '-' * 20,
            '-' * 30,
            '-' * 20
        )
        print(line)
        print(
            '| {:^4} | {:^30} | {:^20} | {:^30} | {:^20} |'.format(
                "№",
                "Фамилия",
                "Имя",
                "Номер телефона",
                "Дата Рождения"
            )
        )
        print(line)

        for idx, contact in enumerate(contacts, 1):
            print(
                '| {:>4} | {:<30} | {:<20} | {:<30} | {:>20} |'.format(
                    idx,
                    contact.get('family', ''),
                    contact.get('name', ''),
                    contact.get('number', 0),
                    '.'.join((str(i) for i in contact['born']))
                )
            )
        print(line)


    elif command.startswith('select'):

        count = 0
        period = input('Введите Фамилию человека, информацию по которому Вы хотите найти: ')

        for contact in contacts:
            if contact.get('family') == period:
                count += 1
                print(
                    '{:>4} {}: {}, {}'.format(contact.get('family', ''),
                                              contact.get('name', ''),
                                              contact.get('number', ''),
                                              '.'.join((str(i) for i in contact['born'])))
                )

        if count == 0:
            print('Человека с такой фамилией в списке нет.')
    elif command == 'help':

        print("Список команд:\n")
        print("add - Добавить контакт;")
        print("list - Вывести список контактов;")
        print("select Поиск по фамилии;")
        print("help - Отобразить справку;")
        print("exit - Завершить работу с программой.")
    else:
        print(f"Неизвестная команда {command}", file=sys.stderr)
```

Рисунок 2 – Задание 1

Ответы на вопросы

1. Что такое словари в языке Python?

Словарь – это изменяемый (как список) неупорядоченный (в отличие от строк, списков и кортежей) набор элементов "ключ: значение".

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

Да, может.

3. Какие методы обхода словарей Вам известны?

Метод перебора элементов в цикле for, методы clear(), copy(), fromkeys(), get(), pop(), popitem(), setdefault(), update(), items(), keys() и values().

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

Методом get(), keys() и values() можно получить значение словаря по ключу.

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

Методом setdefault (), fromkeys() можно получить значение словаря по ключу.

6. Что такое словарь исключений?

Исключения (exceptions) – ещё один тип данных в python. Исключения необходимы для того, чтобы сообщать программисту об ошибках. Самый простой пример исключения – деление на ноль.

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. Пример:

```
employee_numbers = [3, 12, 17, 20]
employee_names = ["Алёна", "Александр", "Иван", "Юля"]


zipped_values = zip(employee_names, employee_numbers)
zipped_list = list(zipped_values)
print(zipped_list)
```

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.

Вывод: Были приобретены и проработаны навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x, а также методы datetime() и zip().