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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.6
дисциплины «Программирование на Python»
Вариант №2

Выполнила:
Беседина Инга Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических
наук, доцент, доцент кафедры
инфокоммуникаций

_____
(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.

Ход работы

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

- *ввод с клавиатуры данных в список, состоящий из заданных словарей;*
- *записи должны быть размещены по алфавиту;*
- *вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;*
- *если таких работников нет, вывести на дисплей соответствующее сообщение.*

```
#!/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? ")
            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)
```

```
>>> help
Список команд:

add - добавить работника;
list - вывести список работников;
select <стаж> - запросить работников со стажем;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Фамилия и инициалы? Журавлёв В. А
Должность? Директор
Год поступления? 1997
>>> add
Фамилия и инициалы? Третьякова М. К
Должность? Дизайнер
Год поступления? 2005
>>> list
+------+----------------------------------+----------------------+----------+
|  №   |             Ф.И.О.               |      Должность       |   Год    |
+------+----------------------------------+----------------------+----------+
|    1 | Журавлёв В. А                    | Директор             |     1997 |
|    2 | Третьякова М. К                  | Дизайнер             |     2005 |
+------+----------------------------------+----------------------+----------+
>>> select 26
   1: Журавлёв В. А
>>> select 3
   1: Журавлёв В. А
   2: Третьякова М. К
>>> select 30
Работники с заданным стажем не найдены.
>>> exit

Process finished with exit code 0
```

Рисунок 1. Результат выполнения программы

9. 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.

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

import sys
```

```python
if __name__ == '__main__':
    school = {}

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

        if command == 'exit':
            break

        # Добавить класс
        elif command == 'add':
            school_class = input('Класс: ').upper()
            students = int(input('Количество учеников: '))
            school.setdefault(school_class, students)

        # изменить количество учеников в классе
        elif command == 'edit':
            school_class = input('Изменить класс: ').upper()
            school[school_class] = int(input('Количество учеников: '))

        # Удалить класс
        elif command == 'delete':
            school_class = input('Удалить класс: ').upper()
            del school[school_class]

        # Вывести данные
        elif command == 'list':
            line = '+-{}-+-{}-+'.format('-' * 4, '-' * 10,)
            print(line)

            print('| {:^4} | {:^10} |'.format("Класс", "Ученики"))
            print(line)

            for key, value in school.items():
                print('| {:>4} | {:<10} |'.format(key, value))
            print(line)

        # Вычислить общее количество учащихся в школе
        elif command == 'total':
            total_students = sum(school.values())
            print("Общее количество учащихся в школе:", total_students)

        # Вывести справку о работе с программой.
        elif command == 'help':
            print("Список команд:\n")
            print("add - добавить новый класс;")
            print("edit - изменить количество учащихся;")
            print("delete - удалить класс;")
            print("total - вывести общее количество учащихся;")
            print("list - вывести список классов;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")
        else:
            print(f"Неизвестная команда {command}", file=sys.stderr)
```

```
>>> help
Список команд:

add - добавить новый класс;
edit - изменить количество учащихся;
delete - удалить класс;
total - вывести общее количество учащихся;
list - вывести список классов;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Класс: 1а
Количество учеников: 23
>>> add
Класс: 1б
Количество учеников: 27
>>> add
Класс: 1в
Количество учеников: 30
>>> list
+------+------------+
| Класс |  Ученики   |
+------+------------+
|   1А | 23         |
|   1Б | 27         |
|   1В | 30         |
+------+------------+
>>> total
Общее количество учащихся в школе: 80
>>> edit
Изменить класс: 1в
Количество учеников: 21
>>> list
+------+------------+
| Класс |  Ученики   |
+------+------------+
|   1А | 23         |
|   1Б | 27         |
|   1В | 21         |
+------+------------+
>>> total
Общее количество учащихся в школе: 71
>>> delete
Удалить класс: 1б
>>> list
+------+------------+
| Класс |  Ученики   |
+------+------------+
|   1А | 23         |
|   1В | 21         |
+------+------------+
>>> total
Общее количество учащихся в школе: 44
>>> exit

Process finished with exit code 0
```

Рисунок 2. Результат выполнения программы

11. Решите задачу: создайте словарь, где ключами являются числа, а значениями – строки. Примените к нему метод *items()*, с с помощью полученного объекта `dict_items` создайте новый словарь, "обратный" исходному, т. е. ключами являются строки, а значениями – числа.

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

if __name__ == '__main__':
    original_dict = {
        1: "one",
        2: "two",
        3: "three",
        4: "four",
        5: "five"
    }

    reversed_dict = {value: key for key, value in original_dict.items()}

    print(reversed_dict)
```

```
{'one': 1, 'two': 2, 'three': 3, 'four': 4, 'five': 5}
```

Рисунок 3. Результат выполнения программы

2. Использовать словарь, содержащий следующие ключи: фамилия и инициалы; номер группы; успеваемость (список из пяти элементов). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в список, состоящий из словарей заданной структуры; записи должны быть упорядочены по возрастанию среднего балла; вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5; если таких студентов нет, вывести соответствующее сообщение.

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


import sys


def display_students(staff):
    """Отобразить список студентов"""
    # Заголовок таблицы.
    line = '+-{}-+-{}-+-{}-+-{}-+'.format(
        '-' * 4,
        '-' * 30,
        '-' * 20,
        '-' * 14
    )
    print(line)
    print(
        '| {:^4} | {:^30} | {:^20} | {:^14} |'.format(
            "№",
            "Ф.И.О.",
            "Группа",
            "Оценки"
        )
    )
    print(line)
```

```
    # Вывести данные о студентах.
    for idx, student in enumerate(staff, 1):
        print(
            '| {:>4} | {:<30} | {:<20} | {:>14} |'.format(
                idx,
                student.get('name', ''),
                student.get('group_number', ''),
                ', '.join(str(el) for el in student.get('grades'))
            )
        )
    print(line)


def main():
    # Список студентов.
    students = []

    # 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':
            surname = input("Введите фамилию и инициалы: ")
            group_num = input("Введите номер группы: ")
            print('Введите оценки: ')
            grades = [int(n) for n in input().split()]

            student = {
                'name': surname,
                'group_number': group_num,
                'grades': grades
            }
            students.append(student)

            # Сортировка по возрастанию среднего балла
            students.sort(key=lambda x: sum(x['grades']) / 5)

        elif command == 'list':
            display_students(students)

        elif command == 'select':
            # Сформировать список студентов, имеющих оценки 4 и 5.
            result = []
            for student in students:
                if all(grade >= 4 for grade in student['grades']):
                    result.append(student)

            display_students(result)

        elif command == 'help':
            # Вывести справку о работе с программой.
            print("Список команд:\n")
            print("add - добавить студента;")
            print("list - вывести список студентов;")
            print("select - запросить студентов, имеющих оценки 4, 5;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")

        else:
```

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


if __name__ == '__main__':
    main()
```

```
>>> help
Список команд:

add - добавить студента;
list - вывести список студентов;
select - запросить студентов, имеющих оценки 4, 5;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Введите фамилию и инициалы: Журавлёв А.В
Введите номер группы: 1
Введите оценки:
5 5 5 4 5
>>> add
Введите фамилию и инициалы: Михайлова С.А
Введите номер группы: 2
Введите оценки:
3 3 4 3 4
>>> add
Введите фамилию и инициалы: Третьяков К.Б
Введите номер группы: 3
Введите оценки:
4 4 4 5 4
>>> list
+------+--------------------------------+----------------------+----------------+
|  №   |             Ф.И.О.             |        Группа        |     Оценки     |
+------+--------------------------------+----------------------+----------------+
|    1 | Михайлова С.А                  | 2                    |  3, 3, 4, 3, 4 |
|    2 | Третьяков К.Б                  | 3                    |  4, 4, 4, 5, 4 |
|    3 | Журавлёв А.В                   | 1                    |  5, 5, 5, 4, 5 |
+------+--------------------------------+----------------------+----------------+
>>> select
+------+--------------------------------+----------------------+----------------+
|  №   |             Ф.И.О.             |        Группа        |     Оценки     |
+------+--------------------------------+----------------------+----------------+
|    1 | Третьяков К.Б                  | 3                    |  4, 4, 4, 5, 4 |
|    2 | Журавлёв А.В                   | 1                    |  5, 5, 5, 4, 5 |
+------+--------------------------------+----------------------+----------------+
>>> exit

Process finished with exit code 0
```

Рисунок 4. Результат выполнения программы

Контрольные вопросы:

1. Словари в Python - это структура данных, которая хранит коллекцию пар ключ-значение, где каждый ключ уникален.

2. Да, функция len() может быть использована для получения количества элементов в словаре.

3. Методы обхода словарей включают использование циклов for для перебора ключей, значений или пар ключ-значение, а также методы keys(), values() и items().

4. Значения из словаря по ключу можно получить с помощью оператора [] или метода get().

5. Значение в словаре по ключу можно установить с помощью оператора [] или метода update().

6. Словарь включений (dictionary comprehensions) - это способ создания нового словаря путем итерации по другой последовательности и определения пар ключ-значение с помощью выражений.

7. Функция zip() используется для объединения элементов из нескольких последовательностей в одну последовательность кортежей.

8. Модуль datetime предоставляет классы и функции для работы с датой и временем в Python. Он позволяет создавать объекты даты, времени, даты и времени, выполнять математические операции с датами, форматировать даты и многое другое.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе со словарями при написании программ с помощью языка программирования Python версии 3.x.