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 ЛАБОРАТОРНОЙ РАБОТЕ №9
дисциплины «Программирование на Python»

Выполнил:
Середа Кирилл Витальевич
1 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Роман Александрович

_____
(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 выполнения:

1) Решить задачу: 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.

```
Исходный словарь:
{'1а': 12, '1б': 32, '2б': 24, '6а': 23, '7в': 30}

Общее количество учащихся в школе: 121

Измененный словарь:
{'1а': 20, '1б': 32, '6а': 23, '7в': 30, '8г': 29}

Общее количество учащихся в школе: 134
```

Рисунок 1 – Результат выполнения программы

Код программы:

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

if __name__ == '__main__':

    school = {
        '1а': 12,
        '1б': 32,
        '2б': 24,
        '6а': 23,
        '7в': 30,
    }

    print("Исходный словарь:")
    print(school)

    total_students = sum(school.values())
    print("\nОбщее количество учащихся в школе:", total_students)

    school['1а'] = 20
```

```
    school['8г'] = 29

    if '2б' in school:
        del school['2б']

    print("\nИзмененный словарь:")
    print(school)

    total_students = sum(school.values())
    print("\nОбщее количество учащихся в школе:", total_students)
```

2) Решить задачу: 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.

```
Исходный словарь:
{1: 'один', 2: 'два', 3: 'три', 4: 'четыре'}

Новый словарь (обратный):
{'один': 1, 'два': 2, 'три': 3, 'четыре': 4}
```

Рисунок 2 – Результат выполнения программы

Код программы:

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

if __name__ == '__main__':

    original_dict = {
        1: 'один',
        2: 'два',
        3: 'три',
        4: 'четыре',
    }

    dict_items_object = original_dict.items()

    reversed_dict = {value: key for key, value in dict_items_object}

    print("Исходный словарь:")
    print(original_dict)

    print("\nНовый словарь (обратный):")
    print(reversed_dict)
```

3) Индивидуальное задание 1 – Вариант 1 - 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 номера группы; вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0; если таких студентов нет, вывести соответствующее сообщение.

```
>>> add
Фамилия и инициалы? Андрей П
Номер группы? 3
Успеваемость (через пробел)? 3.4
>>> add
Фамилия и инициалы? Бочкарева С
Номер группы? 1
Успеваемость (через пробел)? 4.5
>>> list
+-------------------------------+----------------+----------------------+
|            Ф.И.О.             |  Номер группы  |     Успеваемость     |
+-------------------------------+----------------+----------------------+
| Бочкарева С                   | 1              | 4.5                  |
+-------------------------------+----------------+----------------------+
>>> help
Список команд:

add - добавить студента;
list - вывести список студентов;
help - отобразить справку;
exit - завершить работу с программой.
>>>
```

Рисунок 3 – Результат выполнения программы

Код программы:

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

if __name__ == '__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':
            # Запросить данные о студенте.
            full_name = input("Фамилия и инициалы? ")
            group_number = input("Номер группы? ")
            grades_str = input("Успеваемость (через пробел)? ")
```

```
            # Преобразовать строки с оценками в список чисел.
            grades = [float(grade) for grade in grades_str.split()]

            # Создать словарь.
            student = {
                'full_name': full_name,
                'group_number': group_number,
                'grades': grades,
            }

            # Добавить словарь в список.
            students.append(student)

            # Отсортировать список по номеру группы.
            students.sort(key=lambda item: item.get('group_number', ''))

        elif command == 'list':
            # Заголовок таблицы.
            line = '+-{}-+-{}-+-{}-+'.format(
                '-' * 30,
                '-' * 15,
                '-' * 20
            )
            print(line)
            print(
                '| {:^30} | {:^15} | {:^20} |'.format(
                    "Ф.И.О.",
                    "Номер группы",
                    "Успеваемость"
                )
            )
            print(line)
            # Вывести данные о студентах с успеваемостью более 4.0.
            for student in students:
                average_grade = sum(student.get('grades', 0)) /
len(student.get('grades', 1))
                if average_grade > 4.0:
                    print(
                        '| {:<30} | {:<15} | {:<20} |'.format(
                            student.get('full_name', ''),
                            student.get('group_number', ''),
                            ', '.join(map(str, student.get('grades', [])))
                        )
                    )
            print(line)
        elif command == 'help':
            # Вывести справку о работе с программой.
            print("Список команд:\n")
            print("add - добавить студента;")
            print("list - вывести список студентов;")
            print("help - отобразить справку;")
            print("exit - завершить работу с программой.")
        else:
            print(f"Неизвестная команда {command}")
```

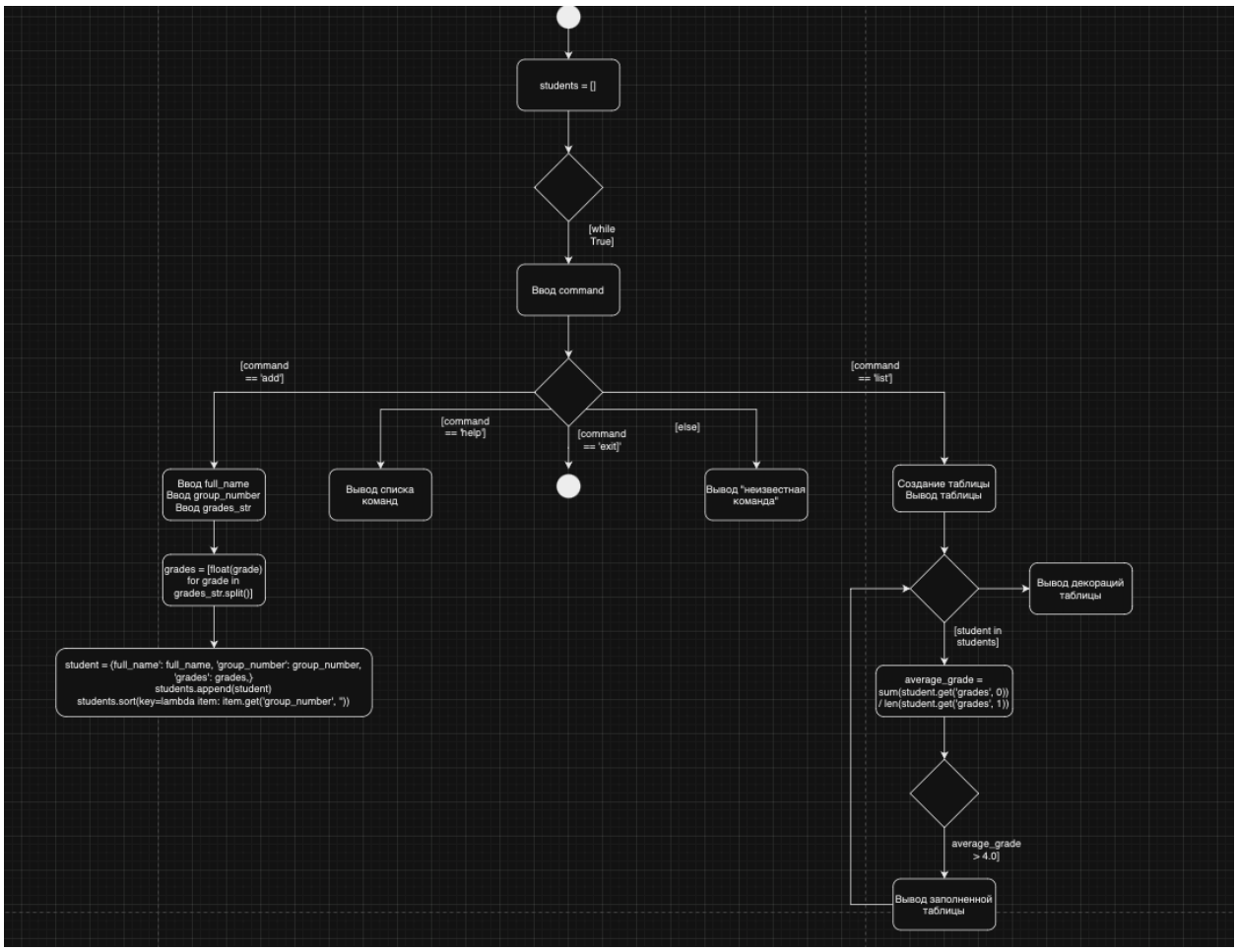

Рисунок 4 – UML диаграмма индивидуального задания

Ответы на 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.

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

Да может, и выведет количество ключей.

3. Какие методы обхода словарей Вам известны?

Цикл for, методы items(), keys() и values().

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

В словаре доступ к значениям осуществляется по ключам, которые заключаются в квадратные скобки или с помощью метода get().

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

С помощью setdefault() можно добавить элемент в словарь. С помощью квадратных скобок: my_dict['a'] = 1. С помощью метода update(): my_dict.update({'a': 1}).

6. Что такое словарь включений?

Словарь включений аналогичен списковым включениям, за исключением того, что он создаёт объект словаря вместо списка (краткая синтаксическая конструкция, предназначенная для создания словаря).

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

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.

Пример:

```
keys = ['a', 'b', 'c']
values = [1, 2, 3]
my_dict = dict(zip(keys, values))
print(my_dict)
```

Данный код выведет: {'a': 1, 'b': 2, 'c': 3}.

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

Модуль datetime в Python позволяет работать с датой и временем, предоставляя классы и функции для работы с текущей датой и временем, форматирования даты и времени, а также вычисления разницы между двумя датами или временем с помощью класса timedelta.

- date — хранит дату
- time — хранит время
- datetime — хранит дату и время

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