**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

**учреждение образования**

**МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ**

**ТЕХНОЛОГИИ**

**Группа 81ТП**

**Отчет**

**по лабораторной работе № 7**

**“**Разработка и отладка алгоритмов работы со словарями. Модуль Json**”**

Учебный предмет

«Инструментальное программное обеспечение»

**Исполнитель: Житкевич М.Д.**

**Руководитель: Бровка Д.С.**

Минск, 2024

Цели:

1. Ознакомление с основами работы со словарями в Python.
2. Изучение понятий сериализации и десериализации данных с использованием модуля json.
3. Развитие практических навыков работы с методами и свойствами словарей и модуля json.

Задачи:

1. Изучить синтаксис и основные операции со словарями, такие как добавление и удаление элементов.
2. Реализовать примеры сериализации словарей в формат JSON и десериализации JSON в словари.
3. Научиться обрабатывать ошибки, возникающие при работе с модулем json.
4. Выполнить практические задачи, включая преобразование данных между словарями и JSON.

Ответы на контрольные вопросы:

1. Словарь (dict) в Python - это структура данных, которая используется для хранения пар ключей и их соответствующих значений. Основная структура словаря - это набор пар ключ-значение, где ключи являются уникальными и неизменяемыми объектами, а значения могут быть любыми объектами.

2. Для создания пустого словаря в Python можно использовать фигурные скобки `{}` или функцию `dict()`. Для добавления элементов в словарь можно просто указать ключ и значение в квадратных скобках. Пример:

my\_dict = {}

my\_dict['key1'] = 'value1'

my\_dict['key2'] = 'value2'

print(my\_dict)

3. Основные методы для работы со словарями в Python:

- `keys()`: возвращает все ключи словаря

- `values()`: возвращает все значения словаря

- `items()`: возвращает все пары ключ-значение

4. Для получения значения по ключу в словаре можно использовать прямое обращение через квадратные скобки: `my\_dict['key']`. Если ключа нет в словаре, то будет возбуждено исключение KeyError. Для избежания ошибки можно использовать метод `get()`, который возвращает значение по ключу, или указанный по умолчанию, если ключ отсутствует.

5. Для перебора всех ключей и значений словаря можно использовать цикл `for` вместе с методом `items()`. Пример:

for key, value in my\_dict.items():

print(key, value)

6. Разница между методом `dict.get()` и прямым обращением по ключу заключается в том, что `get()` возвращает значение по ключу, если он существует, иначе возвращает None или указанное значение по умолчанию. При прямом обращении к элементу словаря по ключу возникает ошибка KeyError, если ключ отсутствует.