ПЗ №8 Работа со словарями в Python

1. Тема работы

Работа со словарями в Python. Создание, модификация и обработка словарей. Использование методов словарей и генераторов словарей.

2. Цель работы

Изучить основы работы со словарями в Python, освоить методы создания и модификации словарей, научиться применять встроенные методы для обработки словарей, изучить генераторы словарей и работу с вложенными структурами.

3. Задачи работы

- Научиться создавать словари различными способами
- Освоить основные методы работы со словарями
- Изучить способы перебора и обработки данных в словарях
- Написать 4 программы для работы со словарями согласно индивидуальному варианту

4. Необходимое оборудование и ПО

- Компьютер с ОС Windows 10
- Установленные: VS Code, Python
- Установленные расширения VS Code: Python, Python Debugger

---

5. Ход работы

Часть 1: Повторение и подготовка

1. **Откройте вашу папку проекта** в VS Code (например, `Иванов_СА-501`)
2. **Создайте новую папку** `Practice_8` внутри вашей основной папки
3. **Создайте файлы** для заданий: `dict1.py`, `dict2.py`, `dict3.py`, `dict4.py`

Часть 2: Разбор примера программы

Пример 1: Базовые операции со словарями

```
# Демонстрация основных операций со словарями
```

```python
# Создание словарей
student = {
    "имя": "Анна",
    "возраст": 20,
    "курс": 3,
    "специальность": "Информатика"
}

grades = {
    "математика": 5,
    "физика": 4,
    "программирование": 5,
    "английский": 4
}

print("Базовые операции со словарями:")
print(f"Студент: {student}")
print(f"Оценки: {grades}")

# Доступ к элементам
print(f"\nДоступ к элементам:")
print(f"Имя студента: {student['имя']}")
print(f"Оценка по математике: {grades.get('математика')}")
print(f"Оценка по химии: {grades.get('химия', 'нет оценки')}")

# Добавление и изменение элементов
student["email"] = "anna@university.edu"
grades["история"] = 4
student["возраст"] = 21

print(f"\nПосле изменений:")
print(f"Студент: {student}")
print(f"Оценки: {grades}")

# Удаление элементов
removed_grade = grades.pop("английский")
print(f"\nУдалённая оценка: {removed_grade}")
print(f"Оценки после удаления: {grades}")
```

Пример 2: Методы словарей и перебор элементов

```
# Демонстрация методов словарей

inventory = {
    "яблоки": 50,
    "бананы": 30,
    "апельсины": 25,
    "груши": 40,
    "киви": 15
}

print(f"Исходный инвентарь: {inventory}")

# Ключи, значения и пары ключ-значение
print(f"\nКлючи: {list(inventory.keys())}")
print(f"Значения: {list(inventory.values())}")
print(f"Пары ключ-значение: {list(inventory.items())}")

# Перебор словаря
print("\nПеребор элементов:")
print("Содержимое склада:")
for product, quantity in inventory.items():
    print(f"  {product}: {quantity} кг")

# Фильтрация
print("\nТовары с количеством больше 30:")
for product, quantity in inventory.items():
    if quantity > 30:
        print(f"  {product}: {quantity} кг")

# Сумма всех товаров
total = sum(inventory.values())
print(f"\nОбщее количество товаров: {total} кг")
```

Пример 3: Вложенные словари и генераторы словарей

```
# Вложенные словари и генераторы словарей

# Вложенный словарь
university = {
    "студент_1": {
        "имя": "Иван",
```

```
        "возраст": 20,
        "оценки": [4, 5, 4, 3]
    },
    "студент_2": {
        "имя": "Мария",
        "возраст": 21,
        "оценки": [5, 5, 4, 5]
    },
    "студент_3": {
        "имя": "Алексей",
        "возраст": 19,
        "оценки": [3, 4, 4, 4]
    }
}

print("Данные студентов:")
for student_id, student_data in university.items():
    avg_grade = sum(student_data["оценки"]) /
len(student_data["оценки"])
    print(f"{student_id}: {student_data['имя']}, возраст:
{student_data['возраст']}, средний балл: {avg_grade:.2f}")

# Генератор словаря
numbers = [1, 2, 3, 4, 5]
squares_dict = {x: x**2 for x in numbers}
print(f"\nКвадраты чисел: {squares_dict}")

# Генератор с условием
even_squares = {x: x**2 for x in numbers if x % 2 == 0}
print(f"Квадраты чётных чисел: {even_squares}")
```

Пошаговый разбор:

1. **Создание словарей** - фигурные скобки `{}` с парами ключ:значение
2. **Доступ к элементам** - по ключу через `[]` или метод `get()`
3. **Добавление элементов** - присваивание значения по новому ключу
4. **Изменение элементов** - присваивание нового значения по существующему ключу
5. **Удаление элементов** - метод `pop()` или `del`
6. **Методы словарей** - `keys()`, `values()`, `items()`
7. **Вложенные словари** - словари внутри словарей

8. **Генераторы словарей** - компактное создание словарей

Запуск программы:

- Сохраните файл (Ctrl+S)
- Запустите через Ctrl+F5 или в терминале:

```
python dict1.py
```

Часть 3: Выполнение задания по варианту

1. **Получите задание** у преподавателя согласно вашему варианту
2. **Реализуйте программы** в соответствующих файлах
3. **Протестируйте программы** на разных входных данных
4. **Убедитесь**, что работа со словарями выполняется корректно

---

6. Варианты заданий

Вариант 1:

1. Напишите программу, которая создаёт словарь с названиями стран и их столицами, затем позволяет пользователю искать столицу по названию страны.
2. Напишите программу, которая инвертирует словарь (меняет местами ключи и значения).
3. Напишите программу, которая объединяет два словаря, при конфликте ключей используя значения из второго словаря.

Вариант 2:

1. Напишите программу, которая создаёт словарь с товарами и их ценами, затем позволяет рассчитывать общую стоимость покупки.
2. Напишите программу, которая инвертирует словарь (меняет местами ключи и значения).
3. Напишите программу, которая находит ключ с максимальным значением в словаре.

Вариант 3:

1. Напишите программу, которая создаёт словарь студентов и их оценок, затем вычисляет средний балл каждого студента.
2. Напишите программу, которая удаляет все элементы с значениями None из словаря.

3. Напишите программу, которая создаёт словарь из двух списков: один список содержит ключи, другой - значения.

Вариант 4:

1. Напишите программу, которая создаёт словарь с месяцами и количеством дней в них, затем позволяет проверять количество дней по названию месяца.
2. Напишите программу, которая сортирует словарь по ключам и по значениям.
3. Напишите программу, которая находит общие ключи в двух словарях.

Вариант 5:

1. Напишите программу, которая создаёт словарь с римскими цифрами и их значениями, затем конвертирует римское число в арабское.
2. Напишите программу, которая создаёт словарь, где ключи - это слова из текста, а значения - количество их вхождений.
3. Напишите программу, которая объединяет несколько словарей в один.

---

7. Отчёт по выполнению работы

Отчёт должен содержать:

1. **Цель работы**
2. **Задание по варианту** (текст всех 3 заданий)
3. **Листинги программ** (код всех 3 программ)
4. **Скриншоты выполнения программ** (результаты работы в терминале)
5. **Выводы**:
   - С какими трудностями столкнулись при работе со словарями?
   - Какие методы словарей оказались наиболее полезными?
   - В чём преимущество словарей перед другими структурами данных?
   - Какая программа была самой сложной и почему?

---

8. Контрольные вопросы

1. Что такое словарь в Python и чем он отличается от списка?
2. Какие требования к ключам словаря? Могут ли ключами быть списки?
3. Как получить значение по ключу? Что произойдёт, если ключа нет?
4. Чем отличается доступ через `dict[key]` от `dict.get(key)`?
5. Как добавить новый элемент в словарь? Как изменить значение существующего элемента?

6. Как удалить элемент из словаря? Чем отличаются `pop()`, `popitem()` и `del`?
7. Как получить все ключи, все значения и все пары ключ-значение?
8. Что такое вложенный словарь и как получить доступ к его элементам?
9. Что такое генератор словаря и в чём его преимущество?
10. Как проверить, существует ли ключ в словаре?

---

9. Рекомендации по выполнению

- Используйте осмысленные имена для словарей (например, `student_grades` вместо `dict1`)
- Для безопасного доступа к значениям используйте метод `get()`
- Используйте генераторы словарей для создания словарей в одну строку
- Помните, что ключи словаря должны быть неизменяемыми типами
- Тестируйте программы на разных наборах данных

---

10. Полезные методы словарей для справки

```
# Основные методы словарей
dict.clear()          # Удалить все элементы
dict.copy()           # Возвращает копию словаря
dict.get(key[, default]) # Получить значение по ключу
dict.items()          # Возвращает пары (ключ, значение)
dict.keys()           # Возвращает ключи
dict.values()         # Возвращает значения
dict.pop(key[, default]) # Удалить ключ и вернуть значение
dict.popitem()        # Удалить и вернуть последнюю пару
dict.update(other)    # Обновить словарь из other
dict.setdefault(key[, default]) # Получить значение, если ключ есть,
иначе установить
```