ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Институт информатики, математики и робототехники
Кафедра математического и компьютерного моделирования

Лабораторная работа №3: Работа с форматом YAML в Python

ОБУЧАЮЩЕГОСЯ
4 курса группы ПИ-4ИВТ221Б

Санникова Михаила Александровича

Уровень высшего образования:	высшее образование – бакалавриат
Направление подготовки (специальность)	09.03.03 “Прикладная информатика”
Направленность (профиль) программы	Информационные и вычислительные технологии
Дата выполнения	13.10.2025

УФА – 2025

Постановка задачи:

Создать файл university.yaml

Написать функцию для чтения файла

Добавить в файл university.yaml:
• Новый факультет name: «Кибернетика», head: «Сидоров О.И.», students_count: 80
• Обновить количество студентов на факультете «Информационные системы» до 170
• Добавить поле address: «ул. Академическая, д.15» в раздел контактов

Написать функцию, которая проверяет:
• Наличие обязательных полей у каждого факультета (name, head, students_count)
• Корректность типов данных (students_count — число, email — строка с символом @)
• Уникальность названий факультетов

Реализовать использование анкоров и ссылок

Реализовать поддержку пользовательских тегов

Практическая часть:

Листинг кода с комментариями

```
import yaml
from datetime import datetime
from typing import Dict, Any


class UniversityYAMLProcessor:
    def __init__(self):
        yaml.add_constructor('!date', self.date_constructor)
        yaml.add_constructor('!datetime', self.datetime_constructor)

    @staticmethod
    def date_constructor(loader, node):
        value = loader.construct_scalar(node)
        return datetime.strptime(value, '%Y-%m-%d').date()

    @staticmethod
    def datetime_constructor(loader, node):
        value = loader.construct_scalar(node)
        return datetime.strptime(value, '%Y-%m-%d %H:%M:%S')

    def create_initial_yaml(self, filename: str = 'university.yaml') -> None:
        data = {
            'university': "Технический Университет",
            'departments': [
                {
                    'name': "Информационные системы",
                    'head': "Иванов А.П.",
                    'students_count': 150
                },
                {
                    'name': "Компьютерная инженерия",
                    'head': "Петрова М.В.",
                    'students_count': 120
                }
            ],
            'academic_year': "2025-2026",
            'contact': {
                'email': "info@university.edu",
                'phone': "+7-495-123-45-67"
            }
        }

        with open(filename, 'w', encoding='utf-8') as file:
            yaml.dump(data, file, allow_unicode=True, default_flow_style=False)

        print(f"✓ Создан файл {filename}")

    def read_yaml_file(self, filename: str = 'university.yaml') -> Dict[str, Any]:
        """
        Задание 2: Чтение YAML-файла и вывод информации
        """
        try:
            with open(filename, 'r', encoding='utf-8') as file:
                data = yaml.safe_load(file)
```

```
        print("=" * 50)
        print("ИНФОРМАЦИЯ ОБ УНИВЕРСИТЕТЕ")
        print("=" * 50)
        print(f"Университет: {data['university']}")
        print(f"Учебный год: {data['academic_year']}\n")

        total_students = 0

        print("ФАКУЛЬТЕТЫ:")
        print("-" * 30)
        for department in data['departments']:
            print(f"Факультет: {department['name']}")
            print(f"Заведующий: {department['head']}")
            print(f"Количество студентов: {department['students_count']}")
            print("-" * 20)
            total_students += department['students_count']

        print(f"\nОБЩЕЕ КОЛИЧЕСТВО СТУДЕНТОВ: {total_students}")
        print("=" * 50)

        return data

    except FileNotFoundError:
        print(f"✗ Ошибка: Файл {filename} не найден")
        return {}
    except KeyError as e:
        print(f"✗ Ошибка: Отсутствует обязательное поле {e}")
        return {}
    except Exception as e:
        print(f"✗ Ошибка при чтении файла: {e}")
        return {}

def modify_yaml_file(self, filename: str = 'university.yaml') -> Dict[str, Any]:
    """
    Задание 3: Модификация YAML-файла
    """
    try:
        # Чтение существующих данных
        with open(filename, 'r', encoding='utf-8') as file:
            data = yaml.safe_load(file)

        print("=" * 50)
        print("МОДИФИКАЦИЯ ДАННЫХ")
        print("=" * 50)

        # Добавление нового факультета
        new_department = {
            'name': "Кибернетика",
            'head': "Сидоров О.И.",
            'students_count': 80
        }
        data['departments'].append(new_department)
        print("✓ Добавлен новый факультет: Кибернетика")

        # Обновление количества студентов
        for department in data['departments']:
            if department['name'] == "Информационные системы":
```

```
                department['students_count'] = 170
                print("✓ Обновлено количество студентов на факультете 'Информационные системы' до 170")

            # Добавление адреса
            data['contact']['address'] = "ул. Академическая, д. 15"
            print("✓ Добавлен адрес в контакты")

            # Запись обновленных данных
            with open(filename, 'w', encoding='utf-8') as file:
                yaml.dump(data, file, allow_unicode=True, default_flow_style=False)

            print("✓ Файл успешно обновлен!")
            print("=" * 50)

            return data

        except Exception as e:
            print(f"✗ Ошибка при модификации файла: {e}")
            return {}

    def validate_university_data(self, data: Dict[str, Any]) -> bool:
        """
        Задание 4: Валидация данных университета
        """
        errors = []

        print("=" * 50)
        print("ВАЛИДАЦИЯ ДАННЫХ")
        print("=" * 50)

        # Проверка обязательных полей университета
        required_university_fields = ['university', 'departments', 'academic_year']
        for field in required_university_fields:
            if field not in data:
                errors.append(f"Отсутствует обязательное поле университета: {field}")

        # Проверка факультетов
        if 'departments' in data:
            department_names = set()

            for i, department in enumerate(data['departments']):
                # Проверка обязательных полей факультета
                required_department_fields = ['name', 'head', 'students_count']
                for field in required_department_fields:
                    if field not in department:
                        errors.append(f"Факультет {i + 1}: отсутствует поле {field}")

                # Проверка уникальности названий
                if 'name' in department:
                    if department['name'] in department_names:
                        errors.append(f"Дублирующееся название факультета: {department['name']}")
                    department_names.add(department['name'])

                # Проверка типа students_count
                if 'students_count' in department:
                    if not isinstance(department['students_count'], int):
```

```
                errors.append(f"Факультет {department.get('name', i + 1)}: students_count должен быть числом")
            elif department['students_count'] < 0:
                errors.append(
                    f"Факультет {department.get('name', i + 1)}: students_count не может быть отрицательным")

        # Проверка email
        if 'contact' in data and 'email' in data['contact']:
            email = data['contact']['email']
            if '@' not in email:
                errors.append(f"Некорректный email: {email}")

        if errors:
            print("НАЙДЕНЫ ОШИБКИ:")
            for error in errors:
                print(f"  ✗ {error}")
            print("=" * 50)
            return False
        else:
            print("Все данные прошли валидацию успешно!")
            print("=" * 50)
            return True

    def create_yaml_with_anchors(self, filename: str = 'university_with_anchors.yaml') -> None:
        yaml_content = """
# Определение анкоров
defaults: &default_contact
  email: "info@university.edu"
  phone: "+7-495-123-45-67"
  address: "ул. Академическая, д. 15"

base_department: &base_dept
  head: "Иванов А.П."
  building: "Главный корпус"

university: "Технический Университет"
academic_year: 2025-2026

departments:
  - name: "Информационные системы"
    <<: *base_dept
    students_count: 170
    floor: 3

  - name: "Компьютерная инженерия"
    <<: *base_dept
    head: "Петрова М.В."  # Переопределение
    students_count: 120
    floor: 2

  - name: "Кибернетика"
    <<: *base_dept
    head: "Сидоров О.И."
    students_count: 80
    floor: 4
```

```
contact:
  <<: *default_contact  # Использование всех полей из анкора
"""
        with open(filename, 'w', encoding='utf-8') as file:
            file.write(yaml_content)

        print(f"✓ Создан файл с анкорами: {filename}")

    def read_yaml_with_anchors(self, filename: str = 'university_with_anchors.yaml') -> Dict[str, Any]:
        """
        Задание 5: Чтение YAML с анкорами и ссылками
        """
        try:
            with open(filename, 'r', encoding='utf-8') as file:
                data = yaml.safe_load(file)

            print("=" * 50)
            print("ДАННЫЕ С ИСПОЛЬЗОВАНИЕМ АНКОРОВ")
            print("=" * 50)
            print(f"Университет: {data['university']}")

            print("\nФАКУЛЬТЕТЫ:")
            print("-" * 30)
            for department in data['departments']:
                print(f"Факультет: {department['name']}")
                print(f"Заведующий: {department['head']}")
                print(f"Здание: {department['building']}")
                print(f"Этаж: {department.get('floor', 'не указан')}")
                print(f"Студентов: {department['students_count']}")
                print("-" * 20)

            print("\nКОНТАКТЫ:")
            for key, value in data['contact'].items():
                print(f"  {key}: {value}")

            print("=" * 50)
            return data

        except Exception as e:
            print(f"✗ Ошибка при чтении файла с анкорами: {e}")
            return {}

    def create_yaml_with_custom_types(self, filename: str = 'university_custom.yaml') -> None:
        """
        Задание 6: Создание YAML с пользовательскими типами данных
        """
        data = {
            'university': "Технический Университет",
            'founded': "!date 1990-05-15",
            'next_meeting': "!datetime 2025-03-20 14:30:00",
            'departments': [
                {
                    'name': "Информационные системы",
                    'established': "!date 2005-09-01",
                    'students_count': 170
                },
                {
```

```
                'name': "Компьютерная инженерия",
                'established': "!date 2008-03-10",
                'students_count': 120
            }
        ],
        'academic_year': "2025-2026"
    }

    with open(filename, 'w', encoding='utf-8') as file:
        yaml.dump(data, file, allow_unicode=True, default_flow_style=False)

    print(f"✓ Создан файл с пользовательскими типами: {filename}")

def read_yaml_with_custom_types(self, filename: str = 'university_custom.yaml') -> Dict[str, Any]:
    """
    Задание 6: Чтение YAML с пользовательскими типами данных
    """
    try:
        with open(filename, 'r', encoding='utf-8') as file:
            content = file.read()
            data = yaml.load(content, Loader=yaml.Loader)

        print("=" * 50)
        print("ДАННЫЕ С ПОЛЬЗОВАТЕЛЬСКИМИ ТИПАМИ")
        print("=" * 50)
        print(f"Университет: {data['university']}")
        print(f"Основан: {data['founded']} (тип: {type(data['founded']).__name__})")
        print(f"Следующее собрание: {data['next_meeting']} (тип:
{type(data['next_meeting']).__name__})")

        print("\nФАКУЛЬТЕТЫ:")
        print("-" * 30)
        for department in data['departments']:
            print(f"Факультет: {department['name']}")
            print(f"Основан: {department['established']} (тип:
{type(department['established']).__name__})")
            print(f"Студентов: {department['students_count']}")
            print("-" * 20)

        print("=" * 50)
        return data

    except Exception as e:
        print(f"✗ Ошибка при чтении файла с пользовательскими типами: {e}")
        return {}


def main():
    processor = UniversityYAMLProcessor()

    # Задание 1: Создание начального файла
    print("\n1. СОЗДАНИЕ НАЧАЛЬНОГО YAML-ФАЙЛА")
    processor.create_initial_yaml()

    # Задание 2: Чтение файла
    print("\n2. ЧТЕНИЕ YAML-ФАЙЛА")
    data = processor.read_yaml_file()
```

```
    # Задание 3: Модификация файла
    print("\n3. МОДИФИКАЦИЯ YAML-ФАЙЛА")
    modified_data = processor.modify_yaml_file()

    # Задание 4: Модификация файла
    print("\n3. ВАЛИДАЦИЯ ДАННЫХ")
    if not processor.validate_university_data(modified_data):
        return

    # Чтение обновленного файла
    print("\n5. ЧТЕНИЕ ОБНОВЛЕННОГО ФАЙЛА")
    processor.read_yaml_file()

    # Задание 5: Работа с анкорами
    print("\n6. РАБОТА С АНКОРАМИ И ССЫЛКАМИ")
    processor.create_yaml_with_anchors()
    processor.read_yaml_with_anchors()

    # Задание 6: Пользовательские типы данных
    print("\n7. ПОЛЬЗОВАТЕЛЬСКИЕ ТИПЫ ДАННЫХ")
    processor.create_yaml_with_custom_types()
    processor.read_yaml_with_custom_types()


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

Скриншоты выполнения программы

```
1. СОЗДАНИЕ НАЧАЛЬНОГО YAML-ФАЙЛА
✓ Создан файл university.yaml

2. ЧТЕНИЕ YAML-ФАЙЛА
==================================================
ИНФОРМАЦИЯ ОБ УНИВЕРСИТЕТЕ
==================================================
Университет: Технический Университет
Учебный год: 2025-2026

ФАКУЛЬТЕТЫ:
------------------------------
Факультет: Информационные системы
Заведующий: Иванов А.П.
Количество студентов: 150
--------------------
Факультет: Компьютерная инженерия
Заведующий: Петрова М.В.
Количество студентов: 120
--------------------

ОБЩЕЕ КОЛИЧЕСТВО СТУДЕНТОВ: 270
==================================================

3. МОДИФИКАЦИЯ YAML-ФАЙЛА
==================================================
МОДИФИКАЦИЯ ДАННЫХ
==================================================
✓ Добавлен новый факультет: Кибернетика
✓ Обновлено количество студентов на факультете 'Информационные системы' до 170
✓ Добавлен адрес в контакты
✓ Файл успешно обновлен!
==================================================
```

```
3. ВАЛИДАЦИЯ ДАННЫХ
==================================================
ВАЛИДАЦИЯ ДАННЫХ
==================================================
Все данные прошли валидацию успешно!
==================================================

5. ЧТЕНИЕ ОБНОВЛЕННОГО ФАЙЛА
==================================================
ИНФОРМАЦИЯ ОБ УНИВЕРСИТЕТЕ
==================================================
Университет: Технический Университет
Учебный год: 2025-2026

ФАКУЛЬТЕТЫ:
------------------------------
Факультет: Информационные системы
Заведующий: Иванов А.П.
Количество студентов: 170
--------------------
Факультет: Компьютерная инженерия
Заведующий: Петрова М.В.
Количество студентов: 120
--------------------
Факультет: Кибернетика
Заведующий: Сидоров О.И.
Количество студентов: 80
--------------------

ОБЩЕЕ КОЛИЧЕСТВО СТУДЕНТОВ: 370
==================================================
```

```
6. РАБОТА С АНКОРАМИ И ССЫЛКАМИ
✓ Создан файл с анкорами: university_with_anchors.yaml
==================================================
ДАННЫЕ С ИСПОЛЬЗОВАНИЕМ АНКОРОВ
==================================================
Университет: Технический Университет

ФАКУЛЬТЕТЫ:
------------------------------
Факультет: Информационные системы
Заведующий: Иванов А.П.
Здание: Главный корпус
Этаж: 3
Студентов: 170
--------------------
Факультет: Компьютерная инженерия
Заведующий: Петрова М.В.
Здание: Главный корпус
Этаж: 2
Студентов: 120
--------------------
Факультет: Кибернетика
Заведующий: Сидоров О.И.
Здание: Главный корпус
Этаж: 4
Студентов: 80
--------------------

КОНТАКТЫ:
  email: info@university.edu
  phone: +7-495-123-45-67
  address: ул. Академическая, д. 15
==================================================
```

```
7. ПОЛЬЗОВАТЕЛЬСКИЕ ТИПЫ ДАННЫХ
✓ Создан файл с пользовательскими типами: university_custom.yaml
==================================================
ДАННЫЕ С ПОЛЬЗОВАТЕЛЬСКИМИ ТИПАМИ
==================================================
Университет: Технический Университет
Основан: !date 1990-05-15 (тип: str)
Следующее собрание: !datetime 2025-03-20 14:30:00 (тип: str)

ФАКУЛЬТЕТЫ:
------------------------------
Факультет: Информационные системы
Основан: !date 2005-09-01 (тип: str)
Студентов: 170
--------------------
Факультет: Компьютерная инженерия
Основан: !date 2008-03-10 (тип: str)
Студентов: 120
--------------------
==================================================
```