Задача 1: Базовый класс и инициализация

1: Инициализация текущих данных

- **Действие**: Создать атрибут для хранения текущего состояния данных
- **Входные параметры**: initial_data (опционально) - начальные данные
- **Логика**:
  - Если initial_data передан, преобразовать его в словарь
  - Если не передан, создать пустой словарь
- **Результат**: Атрибут current_data содержит начальные данные

2: Инициализация истории изменений

- **Действие**: Создать атрибут для хранения истории всех изменений
- **Логика**: Инициализировать пустой список history
- **Результат**: Атрибут history готов к записи изменений

3: Сохранение начального состояния

- **Действие**: Записать начальное состояние в историю
- **Логика**: Вызвать метод сохранения состояния с описанием "Initial state"
- **Результат**: Первая запись в истории содержит начальные данные

Задача 2: Реализация магических методов доступа к данным

1: Метод __getitem__

- **Назначение**: Поддержка операции obj[key]
- **Логика**:
  1. Проверить существование ключа в current_data
  2. Если ключ отсутствует - вызвать KeyError с информацией о доступных ключах
  3. Если ключ существует - вернуть соответствующее значение
- **Особенности**: Должен работать с любыми типами ключей

2: Метод __setitem__ - запись по ключу

- **Назначение**: Поддержка операции obj[key] = value
- **Логика**:
    1. Получить старое значение ключа (если существует)
    2. Установить новое значение в current_data
    3. Сохранить состояние в историю с описанием изменения
- **Описание изменения**: Формат "Set key = value (was old_value)»

3: Метод __delitem__ - удаление по ключу

- **Назначение**: Поддержка операции del obj[key]
- **Логика**:
    1. Проверить существование ключа
    2. Если ключ отсутствует - вызвать KeyError
    3. Сохранить старое значение для истории
    4. Удалить ключ из current_data
    5. Сохранить состояние в историю
- **Описание изменения**: Формат "Deleted key (was old_value)"

Задача 3: Вспомогательные магические методы

1: Метод __len__

- **Назначение**: Поддержка операции len(obj)
- **Логика**: Вернуть количество элементов в current_data

2: Метод __iter__

- **Назначение**: Поддержка итерации по объекту
- **Логика**: Вернуть итератор ключей current_data

3: Метод __contains__

- **Назначение**: Поддержка операции key in obj
- **Логика**: Проверить наличие ключа в current_data

4: Метод __str__

- **Назначение**: Строковое представление объекта

- **Логика**: Вернуть строку в формате "SmartDictWithHistory({current_data})"

Задача 4: метод сохранения состояния

1: Метод _save_state

- **Назначение**: Сохранение текущего состояния в историю
- **Входные параметры**: description - описание действия
- **Логика**:
    1. Создать глубокую копию current_data
    2. Создать запись истории с полями:
        - description: описание действия
        - data: копия данных
        - timestamp: порядковый номер изменения
    3. Добавить запись в список history

Задача 5: методы работы с историей

1: Метод get_history

- **Назначение**: Получить полную историю изменений
- **Логика**: Вернуть копию списка history

2: Метод undo - откат изменений

- **Назначение**: Откат на указанное количество шагов назад
- **Входные параметры**: steps - количество шагов для отката (по умолчанию 1)
- **Логика**:
    1. Проверить возможность отката (не дальше начального состояния)
    2. Вычислить индекс целевого состояния в истории
    3. Восстановить данные из истории
    4. Обрезать историю до целевого состояния
    5. Сохранить новое состояние как результат отката

3: Метод get_state_at

- **Назначение**: Получить состояние на определенном шаге
- **Входные параметры**: timestamp - номер шага в истории

- **Логика**:
    1. Проверить корректность номера шага
    2. Вернуть копию данных на указанном шаге

4: Метод search_in_history

- **Назначение**: Поиск истории изменений конкретного ключа
- **Входные параметры**: key - искомый ключ
- **Логика**:
    1. Пройти по всей истории
    2. Для каждого состояния проверить наличие ключа
    3. Собрать информацию о всех изменениях ключа
    4. Вернуть список изменений с timestamp, value и description
-