Документация классов `NsDb` и `create_table`

Описание классов

Класс `NsDb`

Класс `NsDb` реализует структуру данных для хранения сообщений с возможностью управления уникальностью и позициями сообщений.

Конструктор

```
NsDb:new(input_table, key, is_unique)
```

- **Параметры**:
  - `input_table` (table): Таблица, в которой будут храниться сообщения.
  - `key` (string): Ключ, под которым будет храниться подтаблица сообщений.
  - `is_unique` (boolean): Флаг, указывающий, являются ли сообщения уникальными (true) или могут повторяться (false).
- **Возвращает**: Новый объект класса `NsDb`.

Методы

1. **`add(message)`**

   Добавляет сообщение в таблицу.

   - **Параметры**:
     - `message` (string): Сообщение, которое нужно добавить.
   - **Примечание**: Если класс инициализирован с `is_unique = true`, то сообщения будут уникальными.
2. **`mod_key(change_key, message)`**

   Изменяет значение по указанному ключу.

   - **Параметры**:
     - `change_key` (string): Ключ, значение которого нужно изменить.
     - `message` (string): Новое значение для указанного ключа.

3. **`add_pos(message, position)`**

   Добавляет сообщение по указанной позиции.

   - **Параметры**:
     - `message` (string): Сообщение, которое нужно добавить.
     - `position` (number): Позиция, по которой нужно добавить сообщение.
   - **Примечание**: Если `is_unique = true`, то предыдущее сообщение будет удалено, и индексы будут скорректированы.
4. **`del_pos(message)`**

   Удаляет сообщение по значению или позиции.

   - **Параметры**:
     - `message` (string): Сообщение, которое нужно удалить.
   - **Примечание**: Если `is_unique = true`, то сообщение будет удалено, а индексы будут скорректированы.

Класс `create_table`

Класс `create_table` предназначен для создания и получения таблиц, связанных с определенным ключом.

Конструктор

```
create_table:new(input_table, key)
```

- **Параметры**:
  - `input_table` (table): Таблица, в которой будет храниться подтаблица.
  - `key` (string): Ключ, под которым будет храниться подтаблица.
- **Возвращает**: Новый объект класса `create_table`.

Методы

1. **`get_table()`**

   Получает подтаблицу, соответствующую заданному ключу.

   - **Возвращает**: Подтаблицу, хранящуюся по указанному ключу в `input_table`.

Пример использования

```
-- Создаем основную таблицу для хранения данных
```

```lua
local main_table = {}

-- Создаем объект класса NsDb
local nsdb = NsDb:new(main_table, "messages", true)  -- Уникальные
сообщения

-- Добавляем сообщения
nsdb:add("Hello")
nsdb:add("World")

-- Изменяем ключ
nsdb:mod_key("greeting", "Hi there!")

-- Добавляем сообщение по позиции
nsdb:add_pos("New Message", 1)

-- Удаляем сообщение
nsdb:del_pos("Hello")

-- Создаем объект класса create_table
local ct = create_table:new(main_table, "messages")

-- Получаем таблицу сообщений
local messages_table = ct:get_table()
print(messages_table)  -- Выводит таблицу сообщений
```