def most\_common\_word(self):

"""

Find the most common word in the text.

:return: The most frequent word

"""

Определяется метод most\_common\_word внутри класса (например, класса Text).

Он ищет самое часто встречающееся слово в тексте, который хранится в self.text.

В строке документации (docstring) описывается назначение метода.

python

Копировать код

words = self.text.split()

self.text — это строка, в которой хранится текст.

Метод .split() разбивает строку по пробелам и возвращает список слов.

Например:

python

Копировать код

"hello world hello".split() → ["hello", "world", "hello"]

python

Копировать код

frequency = {}

Создаётся пустой словарь frequency.

В нём будем хранить количество вхождений каждого слова:

ключ — слово

значение — сколько раз оно встретилось

python

Копировать код

for word in words:

frequency[word] = frequency.get(word, 0) + 1

Цикл перебирает все слова из списка words.

frequency.get(word, 0) — ищет слово в словаре:

если слово уже есть → возвращает его текущее количество;

если слова ещё нет → возвращает 0.

К этому числу прибавляется 1 и сохраняется обратно.

Пример:

python

Копировать код

words = ["hello", "world", "hello"]

# после первого слова

{"hello": 1}

# после второго слова

{"hello": 1, "world": 1}

# после третьего слова

{"hello": 2, "world": 1}

python

Копировать код

most\_common = max(frequency, key=frequency.get)

Функция max() ищет максимум.

frequency.get передаётся как ключ для сравнения → значит, max() будет искать слово, у которого самое большое значение (частота).

Например, для словаря {"hello": 2, "world": 1} результат будет "hello".

python

Копировать код

return most\_common

Метод возвращает самое частое слово.

✅ Итог:

Метод most\_common\_word() ищет и возвращает слово, которое встречается чаще всего в тексте.

📌 Пример работы:

python

Копировать код

self.text = "apple banana apple orange apple banana"

→ words = ["apple", "banana", "apple", "orange", "apple", "banana"]

→ frequency = {"apple": 3, "banana": 2, "orange": 1}

→ most\_common = "apple"

def unique\_words(self):

"""

Get a list of unique words in the text.

:return: List of unique words

"""

Определяется метод unique\_words, который возвращает список уникальных слов из текста, хранящегося в self.text.

В docstring написано: метод возвращает список уникальных слов.

python

Копировать код

words = self.text.split()

Берём текст (self.text) и разбиваем его методом .split() на список слов.

По умолчанию .split() разделяет по пробелам и символам перевода строки.

Например:

python

Копировать код

"cat dog cat".split() → ["cat", "dog", "cat"]

python

Копировать код

unique = set(words)

Создаётся множество (set) из списка слов.

Особенность множества: оно хранит только уникальные элементы.

Повторяющиеся слова автоматически убираются.

python

Копировать код

["cat", "dog", "cat"] → {"cat", "dog"}

python

Копировать код

return list(unique)

Множество преобразуется обратно в список.

Возвращается список уникальных слов.

Порядок может быть любым, так как множества не сохраняют порядок.

✅ Итог:

Метод unique\_words() возвращает список слов из текста, но каждое слово встречается только один раз.