File: DESIGN.md *__Developer__: Nalin Yetukuri **Date**: 9/24/2024

MY SPLIT

- Define a method `my_split` with one parameter.
- The parameter of the method is `self` of datatype class object `Sentence`.
- Define an accumulator variable `word`.
- Assign an empty list to the accumulator variable `word`.
- Initialize the instance of the attribute `self.words` to an empty list.
- Use the `for loop` to iterate through the string `self.sentence`
- Check every item in `self.sentence` to see if is a blank space or not. -If the item is not a blank space add the item to the list(accumulator variable) `word`.
- Else check if the list `word` is not empty, use `.join()` to join the characters in the list to form a word (string) and append that word into the list 'self.words' using `.append`.
- Now assign an empty string to the list `word`.
- Come out of the `for loop`. Check if there are items in the list `word`(in case the sentence is ending without a blank space).
- If there are items join the items in the list `word` and append that list to `self.words`.
- Return `self.words`.

MY JOIN

- Define a method `my_join` with one parameter.
- The parameter of the method is `self` of datatype class object `Sentence`.
- Assign the instance of the attribute `self.words` to `self.my_split()` to populate `self.words` with a list of words.
- Define a local variable `wordss`.
- Assign an `empty string` to the local variable `wordss`.
- Use the `for loop` to iterate through the list `self.words`.
- `string` is the loop variable.
- In the `for loop` add each item(string) in the list `self.words` into the empty string `wordss`, add another string in the end with a blank space.
- Assign the string `wordss` to the instance of the variable `self.sentence`.
- Return `self.sentence`

MY INDEX

- Define a method `my_index` with two parameters.
- One parameter of the method is `self` of datatype class object `Sentence`.
- Another parameter is `a_word` of datatype `string`.
- Assign the instance of the attribute `self.words` to `self.my_split()` to populate `self.words` with a list of words.
- Define a local variable `index`.
- Assign the `integer 0` to the local variable `index`.
- Use `for loop` to iterate through the list `self.words` with the iterator variable as `word`.
- If `word` is `a_word` return `index`.
- Increment `index` by `1`.

MY POP

- Define a method `my_pop` with two parameters.
- One parameter of the method is `self` of datatype class object `Sentence`.
- Another parameter is `index` of datatype `int`.
- Assign the instance of the attribute `self.words` to `self.my_split()` to populate `self.words` with a list of words.
- If `index` is greater than the length of the list `self.words` return `None`.
- Define a local variable `word` of datatype `string`.
- Access the element in the list `self.words` at the given `index` and assign it to the local variable `word`.
- Define a local variable `new_list` of datatype `list`.
- Assign an empty list to the local variable `new_list`.
- Use for loop to iterate through the range of length of the list `self.words` with the iterator variable `current_index`.
- If the `current_index` is not equal to `index` then access the element, in the list `self.words` at `current_index`.
- Append the accessed element to `new_list`.
- Come out of the `if` statement.
- Assign `new_list` to `self.words`
- Return `word`.