In this assignment, you will get more practice scanning. In addition, you will learn instance
variables, properties and class instantiation. This work will build on your previous assignment.

**Task**

Write a class that defines two properties: (1) `Input` of type `String`, and (2) `TokenArray`, of type `String[]`. You can assume a maximum size, `MAX_TOKENS`, for `TokenArray`. `Input` is an editable
property and `TokenArray` is a readonly computed property. `Input` is expected to contain digit
sequences (numbers) and letter sequences (words). You can assume that there:

1. is exactly one space character after each number/word (including the last one).
2. is at least one number/word in `Input`.
3. are never more than `MAX_TOKEN` numbers/words in `Input`.
4. are no errors.

Your task is to scan `Input` for the words and numbers and store them in
`TokenArray`. More specifically, if you find N words and numbers in the string, then you
should store them in the first N slots of the array. 1. *You should not convert a digit
sequence into a number for this assignment – you should store it directly as a string.*

You should write two main programs to test this class. One takes a string
argument, prints it, and then uses an instance of this class to set `Input` and then print all
the elements of `TokenArray`. The other takes no string argument, and user ObjectEditor
to interact with an instance of the class. The following screen shots from ObjectEditor
illustrate the interaction with the object.
There are two parts to this assignment. The first is to recognize words in addition to numbers. The second is to put the scanner code into a class with the two properties. It may be best to do the assignment in two parts. First, change the last assignment to recognize and print words and numbers. Next, create an instantiatable class with the required properties. Let me know ASAP if ObjectEditor creates problems – that is does not display or set a property value correctly.

Your displays might not be identical to the ones I show in the assignment handout even when you use the same examples. The difference may be caused by different versions of ObjectEditor (I am using the latest version) and initial values. The displays and initial values do not matter, anything is acceptable. What matters is that you implement and update the properties correctly.
**Constraints**

1. Java has libraries that make the writing of this program trivial. The only library functions you should use are `Character.isDigit()`, `Character.isLetter()`, and the string operations, `charAt()`, `substring()`, and `length()` discussed in class. `Character.isLetter()` is like `Character.isUpperCase()` except that it tells us whether a character is a letter rather than whether it is an uppercase letter. `substring()`, `length()` and `charAt()` are explained in the class material. You do not need to use `Integer.parseInt()` for this assignment.

2. You should write separate methods to search for words and numbers, which can, of course, call some common method.

3. The unfilled slots of the array should be assigned to the empty string (""). This means all of the array slots must be initialized by the time first call to the getter for `TokenArray` returns a value. *If you do not do that, the released version of ObjectEditor will throw an exception.*

4. The `Input` property should also be initialized to the empty string. If you do not do that, `ObjectEditor` cannot let you edit the value of this property.

5. As always, use meaningful variable names to make your code understandable to the reader. Comment if you think it is necessary.

**Extra Credit**

1. Allow an arbitrary number of spaces before and after each number/word.

2. Give error messages in case the user does not enter any number/word or enters non digits/letters.

3. Look also for a token consisting of a single character, the double quote ““” and store that also in `TokenArray`.

4. Add another property to the class so that the main program can only the tokens you found rather than the whole `TokenArray`.

**Submission Instructions**

The submission instructions for this assignment:

1. Submit a print out of your code at the start of class together with Figure 1 like screen shots showing your code working. These screen shots should show the output of the first main
program, and interaction with ObjectEditor for the second main program. To print a window, do the following. Click on a window and then press Alt and PrtSc keys together. This will copy the image to a buffer. You can now paste the buffer in a document.

2. Upload the assignment directory in blackboard. In general, for all assignments, you should do so by midnight of the day the assignment is due. But do not change the code after you submit it in class.

Good luck!