BRAC University

Department of Computer Science and Engineering

CSE 220: Data Structures

Assignment 01

Task 1

Implement a **MyString** ADT.

Elements

An empty array of characters.

Structure of the Elements

A collection of characters. The characters in a string are in sequential (or linear) order – that is, the characters follow one after the other from the beginning of a string to its end. The character positions are numbered beginning with zero. A word, phrase, or sentence is some examples of strings.

The characters in the instance of MyString will be stored in an array.

CONSTRUCTORS

MyString ( )

Precondition:

None.

Postcondition:

This is the default constructor. It creates an empty MyString object (just a MyString reference).

Example:

```
... main ( ){

...

MyString a = new MyString( );

...

}
```

MyString (char[ ] charSeq)

Precondition:

An array of characters charSeq will be given to create the constructor.

Postcondition:

It creates a new MyString object with a character sequence identical to the character array charSeq.

Example:

```
... main ( ){
```

...

MyString a = new MyString(c); // c is a character array

...

}

MyString (String str)

Precondition:

A String str will be given to create the constructor.

Postcondition:

This creates a new MyString object whose contents are equivalent to the String str.

Example:

... main ( ){

...

MyString a = new MyString("cat");

...

}

METHODS

[Some of the more commonly used String class methods-but you CANNOT use the String class methods here. You have to implement these methods on your own.]

int length ( )

Precondition:

None.

Postcondition:

Returns the number of characters in the MyString object.

char charAt (int n)

Precondition:

"*n*" must be a valid String index (which is an integer) less than the length of the MyString object where you invoke this method (check the validity for *n*, e. g., *n* is an integer, non-negative and less than the length of the String).

Postcondition:

Returns the n^{th} character in the MySrting object.

boolean startsWith (MyString prefix)

Precondition:

A MyString object *prefix* that is not null.

Postcondition:

Returns true if the MyString object starts with "*prefix*". Otherwise, returns false.

boolean startsWith (String prefix)

Precondition:

A String object *prefix* that is not null.

Postcondition:

Returns true if the MyString object starts with "*prefix*". Otherwise, returns false.

boolean endsWith(MyString suffix)

Precondition:

A MyString object *suffix* that is not null.

Postcondition:

Returns true if the MyString object ends with "*suffix*". Otherwise, returns false.

boolean endsWith(String suffix)

Precondition:

A String object *suffix* that is not null.

Postcondition:

Returns true if the MyString object ends with "*suffix*". Otherwise, returns false.

MyString replaceFirst(char oldChar, char newChar)

Precondition:

Two valid characters “*oldChar*” and “*newChar*”.

Postcondition:

Returns a new MyString resulting from replacing the first occurrence of the *oldChar* in this string with the *newChar*.

MyString replaceAll(char oldChar, char newChar)

Precondition:

Two valid characters “*oldChar*” and “*newChar*”.

Postcondition:

Returns a new MyString resulting from replacing all occurrences of the *oldChar* in this string with the *newChar*.

MyString replaceLast(char oldChar, char newChar)

Precondition:

Two valid characters “*oldChar*” and “*newChar*”.

Postcondition:

Returns a new MyString resulting from replacing the last occurrence of the *oldChar* in this string with the *newChar*.

MyString toLowerCase ( )

Precondition:

None.

Postcondition:

Returns the invoking MyString object if all its characters are already lowercase. Otherwise, returns a new MyString object in which all characters have been converted to lowercase.

MyString toUpperCase ( )

Precondition:

None.

Postcondition:

Returns the invoking MyString object if all its characters are already uppercase. Otherwise, returns a new String object in which all characters have been converted to uppercase.

boolean equals (MyString rightStr)

Precondition:

A MyString object *rightStr* and *rightStr* is not null. (check validity of *rightStr*)

Postcondition:

It returns true if the invoking MyString object and rightStr have the same value (i.e, identical). Otherwise, returns false.

boolean equalsIgnoreCase (MyString rightString)

Precondition:

A MyString object *rightStr* and *rightStr* is not null. (check validity of *rightStr*)

Postcondition:

It returns true if the invoking MyString object and *rightString* are identical not considering the case (uppercase or lowercase) to each character. Otherwise, returns false.

int compareTo (MyString str)

Precondition:

A MyString object *str*. *Str* is not null. (check validity of *str*)

Postcondition:

Returns a value indicating if the invoking MyString object is lexicographically before (returns a negative value), equal to (returns 0), or after (returns a positive value) the MyString str.

Example:

```
... main ( ){

...

a.MyString(b); // a and b are instances of MyString Class

...

}
```

[This method returns 0 if a and b are identical, returns negative value if a<b and returns positive value if a > b.]

int compareTo (String str)

Precondition:

A String object *str* and *Str* is not null. (check validity of *str*)

Postcondition:

Returns a value indicating if the invoking MyString object is lexicographically before (returns a negative value), equal to (returns 0), or after (returns a positive value) the String str.

Example:

... main ( ){

...

a.MyString("book"); // a is an instances of MyString Class

...

}

int compareToIgnoreCase(MyString str)

Precondition:

A MyString object *str* where *str* not null (check validity of *str*).

Postcondition:

Returns a value indicating if the invoking MyString object is lexicographically before (returns a negative value), equal to (returns 0), or after (returns a positive value) the MyString *str*, if both are not case sensitive.

int compareToIgnoreCase(String str)

Precondition:

A String object *str* where *str* not null (check validity of *str*).

Postcondition:

Returns a value indicating if the invoking MyString object is lexicographically before (returns a negative value), equal to (returns 0), or after (returns a positive value) the String *str*, if both are not case sensitive.

MyString substring (int start)

Precondition:

The argument "*start*" must be a nonnegative String index and is not greater than the length of the MyString object.

Postcondition:

Returns a new MyString object containing the substring from the index "*start*" to the end of the invioking MyString object.

MyString substring (int start, int end)

Precondition:

The "*start*" and "*end*" must be nonnegative String indices and are not greater than the length of the MyString. Moreover, "*start"* must not be greater than "*end"*. (check validity)

Postcondition:

Returns a new MyString object containing the substring starting at position "*start*" through position "*end*" of the invoking MyString object. [Here, a total of "*end* – *start* + 1" characters are copied into the new MyString object].

int indexOf (char ch)

Precondition:

A character "*ch*", where "*ch*" is not null. (check validity of "ch")

Postcondition:

Returns the position within the invoking MyString object at which the first (the leftmost) occurrence of the character "*ch*" is located. If "*ch*" is not found, -1 is returned.

int lastIndexOf (char ch)

Precondition:

A character "*ch*", where "*ch*" is not null. (check validity of "*ch*")

Postcondition:

Returns the index within this Mystring object of the last (rightmost) occurrence of the specified character "ch". If "ch" is not found, -1 is returned.

int indexOf (char ch, int start)

Precondition:

A character “*ch*”, where “*ch*” is not null. (check validity of “*ch*”) and the starting position “*start*” to start searching within the MyString object.

Postcondition:

Returns the position within the invoking MyString object at which the first (the leftmost) occurrence of the character “*ch*” is located, with “start” specifying the position at which to begin the search. If “*ch*” is not found, then -1 is returned.

int lastIndexOf (char ch, int start)

Precondition:

A character “*ch*”, where “*ch*” is not null. (check validity of “*ch*”) and the starting position “*start*” to start searching within the MyString object.

Postcondition:

Returns the index within this string of the last occurrence of the specified character, searching from the position “*start*”. If “*ch*” is not found, -1 is returned.

int indexOf (MyString str)

Precondition:

A MyStrjng object *str* that is not null. (check validity of *str*)

Postcondition:

Returns the position within the invoking MyString object at which the first (the leftmost) occurrence of the MyString str is located. If “*str*” is not found, then -1 is returned.

int lastIndexOf (MyString str)

Precondition:

A MyStrjng object *str* that is not null. (check validity of *str*)

Postcondition:

Returns the position within the invoking MyString object at which the last (the righttmost) occurrence of the MyString str is located. If "*str*" is not found, -1 is returned.

int indexOf (String str)

Precondition:

A Strjng object *str* that is not null. (check validity of *str*)

Postcondition:

Returns the position within the invoking MyString object at which the first (the leftmost) occurrence of the String *str* is located. If "*str*" is not found, then -1 is returned.

int lastIndexOf (String str)

Precondition:

A Strjng object *str* that is not null. (check validity of *str*)

Postcondition:

Returns the position within the invoking MyString object at which the last (the rightmost) occurrence of the String *str* is located. If "*str*" is not found, -1 is returned.

MyString concat (MyString str)

Precondition:

A MyString object *str*. The object *str* is not null (check validity of *str*).

Postcondition:

This Returns a new MyString object that containins the MyString object that invoked this method with *str*, added to it at the end.

Example:

string1.concat(string2);

// string 1 and string2 are instances of MyString class.

MyString concat (char[ ] charSeq)

Precondition:

A character array.

Postcondition:

This Returns a new MyString object that containins the MyString object that invoked this method with *charSeq*, added to it at the end.

MyString concat (String str)

Precondition:

A String *str*. The object *str* is not null. (check validity of *str*)

Postcondition:

This Returns a new MyString object that containins the MyString object that invoked this method with a string *str*, added to it at the end.