Strings

zyBook Chap 1.5, 2.9, 4.14, 4.15, 4.16

String class

- The `String` class represents character strings.
- `String` objects are immutable and cannot change.
- `String` objects can be created and assigned like primitive values:
  - **`String <name> = "<text>";`**
  - **`String <name> = <expression>;`**

- For example:

```
String department = "CS";
int courseNum = 1101;
String course = department + courseNum;  // "CS1101"
```

String Indexing

- String objects consist of a list of characters (char)
- Characters are numbered internally with an index
- **Index starts at 0**
- For example,

```
String school = "Vandy";
```

0	1	2	3	4
V	a	n	d	y

How to determine if two String objects are equal?

public boolean equals(Object anObject)

- Parameter: anObject – The object to compare this String against
- Returns: **true** if the given object represents the same sequence of characters as this String, **false** otherwise
- For example,

```
String school = "Vandy";
school.equals("Vandy");       // true
school.equals("Vanderbilt");  // false
```

How many characters are in the String?

`public int length()`

- Returns: the length of the sequence of characters represented by this object.
- For example,

```
String school = "Vandy";
int numLetter = school.length();  // 5
```

0	1	2	3	4
V	a	n	d	y

What is the N-th character of the String?

`public char charAt(int index)`

- Parameter: index – the index of the char value.
- Returns: the char value at the specified index of this string.
- Throws: IndexOutOfBoundsException if the index argument is negative or not less than the length of this string.
- For example,

```
String school = "Vandy";
school.charAt(0);  // 'V'
school.charAt( school.length() – 1 );  // 'y'
school.charAt( school.length() );      // IndexOutOfBoundsException
```

0	1	2	3	4
V	a	n	d	y

Where is a substring in the String?

`public int indexOf(String str)`

- Parameter: str – the substring to search for.
- Returns: the index of the first occurrence of the specified substring, or -1 if there is no such occurrence.
- For example,

```
String school = "Vandy";
school.indexOf("n");    // 2
school.indexOf("an");   // 1
school.indexOf("ad");   // -1
```

0	1	2	3	4
V	a	n	d	y

Substrings

`public String substring(int beginIndex)`

- Parameter: **beginIndex** – the beginning index, **inclusive**.
- Returns: the specified substring.
- Throws: IndexOutOfBoundsException if beginIndex is negative or larger than the length of this String object.
- For example,

```
String school = "Vandy";
school.substring(1);  // "andy"
```

0	1	2	3	4
V	a	n	d	y

`public String substring(int beginIndex, int endIndex)`

- Parameters:
  beginIndex – the beginning index, **inclusive**.
  endIndex – the ending index, **exclusive**.
- Returns: the specified substring.
- Throws: IndexOutOfBoundsException
  - the beginIndex is negative, or
  - the endIndex is larger than the length of this String object, or
  - the beginIndex is larger than endIndex.
- For example,

```
String school = "Vandy";
school.substring(0, 2);  // "Va"
```

0	1	2	3	4
V	a	n	d	y

Uppercase and Lowercase

`public String toLowerCase()`

- Returns: the String, converted to lowercase.

`public String toUpperCase()`

- Returns: the String, converted to uppercase.

- For example,

```
String school = "Vandy";
String lower = school.toLowerCase();   // "vandy"
String upper = school.toUpperCase();   // "VANDY"
```

How to compare two String objects?

`public int compareTo(String anotherString)`

- Parameter: anotherString – the String to be compared.
- Returns:
  - the value 0 if the argument string is equal to this string;
  - a value less than 0 if this string is lexicographically less than the string argument;
  - a value greater than 0 if this string is lexicographically greater than the string argument.
- For example,

```
String school = "Vandy";
school.compareTo("Vandy");  // 0
school.compareTo("VANDY");  // 32
school.compareTo("vandy");  // -32
```

Q: Find the exact output of the following code

```
public class StringExample {

    public static void main(String[] args) {
        String question = "How are you?";
        String response = "I am fine. Thanks.";

        System.out.println(question.length());   // 12
        System.out.println(response.length());   // 18
        System.out.println(question.length() + response.length()); // 30

        String sub1 = question.substring(3, 7);
        System.out.println(sub1.toUpperCase());  //  ARE

        String sub2 = response.substring(7);
        System.out.println(sub2.toLowerCase());  // ne. thanks.
    }
}
```

Reading String Token(s) from Console

- **`public String next()`**
  - Reads and returns user input as a String

- **`public String nextLine()`**
  - Reads and returns and entire line of user input as a String