Tutorial of Constructor and String

> Based on the tutorial of "2020S-Java-A" designed by teaching group in SUSTech
>
> Modified (only change to markdown file) by ZHU Yueming in 2021. April. 6th
>
> Update by ZHU Yueming in 2021. Nov. 1st. Add File read and write demo

Objective

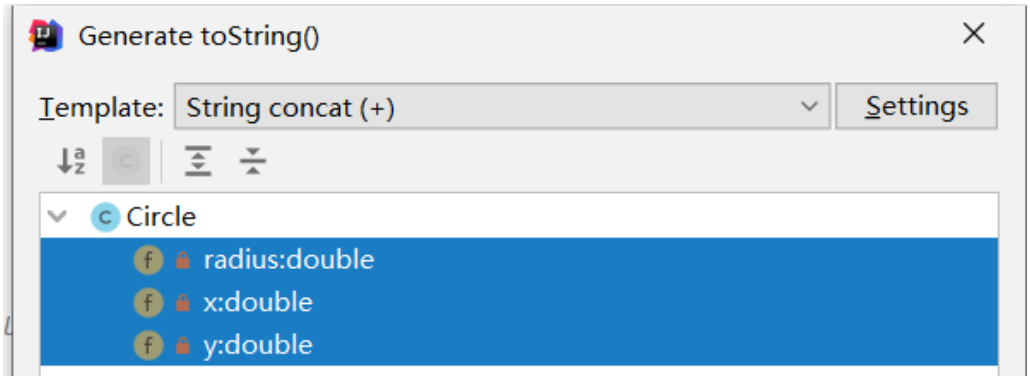
- Learn to declare and use of `toString( )` method.
- Learn to use various String methods and StringBuilder

Part 1: toString( ) method

A special instance method **`toString( )`** which return a string related to the current object.

While the object is used as a string, the `toString( )` is invoked by default. For example:

- Create a Circle object c : `Circle c = new Circle(1.0, 1.0, 1.0);`
- Use `System.out.print(c)` to print the c, what's the output?
- Use `System.out.print(c.toString())` instead of `System.out.print(c)`, what's the output?
- Use IDEA to generated the `toString()` of the current class.



The following instance method `toString()` is generated, all the instance data field is added into the String:

```
public String toString() {
      return "Circle{" +
            "radius=" + radius +
            ", x=" + x +
            ", y=" + y +
            '}';
       }
```

- Invoke `System.out.print(c)` and `System.out.print(c.toString())` again, what's the output?
- While change the `toString( )` as follow code, invoke `System.out.print(c)`. what's the output, why?

```
public String toString(char z) {
      return "Circle{" +
            "radius=" + radius +
            ", x=" + x +
            ", y=" + y +
      ", z=" + z +
            '}';
       }
```

Part 2: String manipulations

Please use String methods

(https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) to finish the tasks below. Methods in the Character class are also helpful:

https://docs.oracle.com/javase/8/docs/api/java/lang/Character.html.

Exercise 1

Write a program to check if a string provided by a user is a palindrome or not. A string is a palindrome is the reverse of the string is the same as the string (we do not differentiate upper-case and lower-case letters in this task). For example, "abba" is a palindrome. "#Aa#" and "0" are also palindromes.

Your program should continuously take user inputs for checking and stops when the user types "quit".

A sample run:

```
Type a string ("quit" to finish): hello
hello is not a palindrome
Type a string ("quit" to finish): many
many is not a palindrome
Type a string ("quit" to finish): 0
0 is a palindrome
Type a string ("quit" to finish): 900
900 is not a palindrome
Type a string ("quit" to finish): #Aa#
#Aa# is a palindrome
Type a string ("quit" to finish): quit

Process finished with exit code 0
```

Exercise 2

Write a program to remove all repeated characters in a string provided by the user and return a new string without any repeating characters or white spaces. Please use `StringBuilder` to build the new string.

Sample runs:

```
Please type a string: hello
After removing repeating chars and spaces: helo
```

```
Please type a string:
Empty string, exit…
```

```
Please type a string: abcd  bcde cdef
After removing repeating chars and spaces: abcdef
```

Exercise 3

Write a program to count the occurrence of a substring in a string. The program should ask the user to input two strings s1 and s2, and output the number of occurrences of s2 in s1.

Sample runs:

```
s1: JavaExamplesJavaCodeJavaProgram
s2: Java
Found at index: 0
Found at index: 12
Found at index: 20
Total occurrences: 3

s1: abcd  bcde cdef
s2: bc
Found at index: 1
Found at index: 6
Total occurrences: 2

s1: abcdefg
s2: xyz
Total occurrences: 0
```

Exercise 4

Input a number n, which means there are n lines, then write a program to print strings in lines as follows. And then improve your program to be more efficient.

```
please input number n:10
a
ab
abc
abcd
abcde
abcdef
abcdefg
abcdefgh
abcdefghi
abcdefghij
```

The following code can resolve the problem, but it is a time-consuming way. Using `StringBuilder` to improve it.

```
Scanner in = new Scanner(System.in);
System.out.print("please input number n:");
int n = in.nextInt();
String str = "";
for (int i = 0; i < n; i++) {
     str = str + (char) (97 + i);// str = new String("abc"+"d"); time-consuming
     System.out.println(str);
}
```

Hint:

```
StringBuilder sb = new StringBuilder();
sb.append('a');
```

API References:

String methods:

https://docs.oracle.com/javase/8/docs/api/java/lang/String.html

```
public int length()

public char charAt(int index)

public boolean startsWith(String prefix)
```

```
public boolean equals(Object anObject)

public boolean equalsIgnoreCase(String anotherString)

public String trim()

public int indexOf(String str)

public int indexOf(String str, int fromIndex)

public String substring(int beginIndex)

public String substring(int beginIndex, int endIndex)

public String[] split(String regex)

public char[] toCharArray()
```

Character methods:

https://docs.oracle.com/javase/8/docs/api/java/lang/Character.html

```
public static char toLowerCase(char ch)

public static boolean isWhitespace(char ch)
```

StringBuilder methods:

https://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html

```
public StringBuilder append(char c)

public String toString()
```