CS102A Introduction to Computer Programming
Fall 2020 Lab 8

Objective

1. Learn to declare constructors and use them to construct objects.

2. Learn to use various String methods.

1 Before Exercise

1.1 Step 1: Constructors and instance methods

The Circle class defined in the last lab does not contain any explicitly declared constructor. The Java compiler will provide a default constructor that would initialize all three fields (radius, x, y) to 0.0 when called. If we want to create a circle object, of which the three fields have the following values: radius = 2.0, x = 1.0, y = 1.0, we can write a main method like the one below.

```java
public static void main(String[] args) {
    Circle c = new Circle();
    c.setRadius(2.0);
    c.setX(1.0);
    c.setY(1.0);
}
```

However, this is quite troublesome. A better solution is to declare constructors so that they can be called to construct objects with certain radius values and center positions. The following code declares two such constructors. The first constructor takes one argument to initialize the radius field (the fields x and y will be initialized to 0.0). The second constructor takes three arguments to initialize all three fields.

```java
public Circle(double radius) {
    this.radius = radius;
```

```
3  }
4
5  public Circle(double radius, double x, double y) {
6      this.radius = radius;
7      this.x = x;
8      this.y = y;
9  }
```

Note that in the constructors, `this` keyword is needed to differentiate the field access from method argument access. Now we can simply create the circle (`radius = 2.0`, `x = 1.0`, `y = 1.0`) with a constructor call. Much easier, right?

```
1  public static void main(String[] args) {
2      Circle c = new Circle(2.0, 1.0, 1.0);
3  }
```

Continue to type the following code in the main method and see what happens.

```
1  Circle c = new Circle();
```

The code would not compile. Do you know why? If not, please go check our lecture notes.

2 Exercise

2.1 Exercise 1

Add a public method `distanceToOrigin()` to the `Circle` class. The method returns the distance between the circle's center point and the origin point (`0.0, 0.0`). Then write a Java program `Lab9E1.java` that performs the following tasks:

1. Generate a random number N in the range $[5, 10)$.

2. Create N circles. Each circle has a random radius in the range $[1.0, 3.0)$ and a random center position: `x` and `y` are in the range $[2.0, 5.0)$.

3. Among the generated circles, find the one with the smallest area and the one whose center is the furthest from the origin point.

> **Tip**
>
> For random number generation, you may use the following two methods of the `Random` class:
>
> ```
> 1  public int nextInt(int bound)
> ```

```java
1 public double nextDouble()
```

See https://docs.oracle.com/javase/8/docs/api/java/util/Random.html for more details.

Sample input and output:

```
Circle #1: radius = 1.71, x = 4.84, y = 4.46
Circle #2: radius = 2.90, x = 2.78, y = 4.08
Circle #3: radius = 2.63, x = 4.29, y = 2.63
Circle #4: radius = 1.24, x = 2.17, y = 4.85
Circle #5: radius = 2.14, x = 3.00, y = 4.21
Circle #6: radius = 2.92, x = 4.36, y = 2.39
Circle #7: radius = 2.55, x = 4.57, y = 4.87
Circle #8: radius = 2.59, x = 2.14, y = 2.12
Circle #9: radius = 2.02, x = 3.40, y = 3.04
Circle #4 is the smallest circle, area = 4.87
Circle #7 is the farthest circle, distance to origin = 6.68
```

2.2 Exercise 2

> **💡Tip**
>
> 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.

Write a program Lab9E2.java 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*.

Sample input and output:

```
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
```

2.3 Exercise 3

Write a program Lab9E3.java 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 input and output:

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

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

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

2.4 Exercise 4

Write a program Lab9E4.java 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 input and output:

```
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
```

💡 **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()`