Chapter 9 - Review Questions

Section 9.2

9.1 Suppose that s1, s2, s3, and s4 are four strings, given as follows:

```
String s1 = "Welcome to Java";
String s2 = s1;
String s3 = new String("Welcome to Java");
String s4 = "Welcome to Java";
```

1. **What are the results of the following expressions?
2. s1 == s2
3. s2 == s3
4. s1.equals(s2)
5. s2.equals(s3)
6. s1.compareTo(s2)
7. s2.compareTo(s3)
8. s1 == s4
9. s1.charAt(0)
10. s1.indexOf('j')
11. s1.indexOf("to")
12. s1.lastIndexOf('a')
13. s1.lastIndexOf("o", 15)
14. s1.length()
15. s1.substring(5)

16. s1.substring(5, 11)

17. s1.startsWith("Wel")

18. s1.endsWith("Java")

19. s1.toLowerCase()

20. s1.toUpperCase()

21. " Welcome ".trim()

22. s1.replace('o', 'T')

23. s1.replaceAll("o", "T")

24. s1.replaceFirst("o", "T")

25. s1.toCharArray()

To create a string "Welcome to Java", you may use a statement like this:
`String s = "Welcome to Java";` or `String s = new String("Welcome to Java);`
2. Which one is better? Why?

Solution

1. Respectively output of expressions at the above.

    1. true
    2. false
    3. true
    4. true
    5. 0
    6. 0
    7. true
    8. W
    9. -1
    10. 8
    11. 14
    12. 9
    13. 15
    14. me to Java
    15. me to
    16. true

    17. true
    18. welcome to java
    19. WELCOME TO JAVA
    20. Welcome
    21. WelcTme tT Java
    22. WelcTme tT Java
    23. WelcTme to Java
    24. Welcome to Java

2. First one, because second one create a new string instead of using *interned string* that created by compiler if same literal used before.

9.2 Suppose that s1 and s2 are two strings. Which of the following statements or expressions are incorrect?

1. String s = new String("new string");
2. String s3 = s1 + s2;
3. String s3 = s1 - s2;
4. s1 == s2;
5. s1 >= s2;
6. s1.compareTo(s2);
7. int i = s1.length();
8. char c = s1(0);
9. char c = s1.charAt(s1.length());

Solution

Respectively output of expressions at the above.

1. Correct
2. Correct
3. Incorrect
4. Correct
5. Incorrect
6. Correct
7. Correct
8. Incorrect
9. Incorrect

9.3 What is the printout of the following code?

```
String s1 = "Welcome to Java";
String s2 = s1.replace("o", "abc");
System.out.println(s1);
System.out.println(s2);
```

Solution

Output

```
Welcome to Java
Welcabcme tabc Java
```

9.4 Let s1 be " Welcome " and s2 be " welcome ". Write the code for the following statements:

Due of the longness, I didn't write statements here.

Solution

```
String s1 = "Welcome ";
String s2 = " welcome ";

boolean isEqual = s1.equals( s2 ); // 1
System.out.println( isEqual );
isEqual = s1.equalsIgnoreCase( s2 ); // 2
System.out.println( isEqual );
int x = s1.compareTo( s2 );
System.out.println( x );
int x = s2.compareToIgnoreCase( s2 );
System.out.println( x );
boolean b = s1.startsWith( "AAA" );
System.out.println( b );
boolean b = s1.endsWith( "AAA" );
System.out.println( b );
int x = s1.length();
System.out.println( x );
char x = s1.charAt( 0 );
System.out.println( x );
String s3 = s1.concat( s2 );
System.out.println( s3 );
String substringExample = s1.substring( 1 );
System.out.println( substringExample );
String substringExample = s1.substring( 1, 4 );
System.out.println( substringExample );
String s3 = s1.toLowerCase();
System.out.println( s3 );
String s3 = s1.toUpperCase();
System.out.println( s3 );
String s3 = s1.trim( s1 );
System.out.println( s3 );
String s3 = s1.replace( E, ' ' );
System.out.println( s3 );
String[] splitted_words = s1.split( " " );
for( int i = 0; i < splitted_words.length; i++ )
        System.out.println( splitted_words[ i ] );
int x = s1.indexOf( 'e' );
System.out.println( x );
int x = s1.lastIndexOf( "abc" );
System.out.println( x );
```

9.5 Does any method in the String class change the contents of the string?

Solution

No, because string is immutable. It can't be change after created.

9.6 Suppose string s is created using new String(); what is s.length()?

Solution

Zero. Because there is no literal input to create.

9.7 How do you convert a char, an array of characters, or a number to a string?

Solution

We can convert char, char array, double, float, int, long and boolean to String with valueOf method.

9.8 Why does the following code cause a NullPointerException?

```
public class Test {
        private String text;

        public Test(String s)
        {
                String text = s;
        }

        public static void main(String[] args)
        {
                Test test = new Test("ABC");
                System.out.println(test.text.toLowerCase());
        }
}
```

Solution

Because in constructor another String variable created as name as text. And parameter of constructor assigned that local variable, not the global variable. Therefore, when toLowerCase method called, compiler gives NullReferenceException because nothing asigned to text the global variable.

9.9 What is wrong in the following program?

```
public class Test
{
        String text;
        public void Test(String s)
        {
                this.text = s;
        }

        public static void main(String[] args)
        {
                Test test = new Test("ABC");

                System.out.println(test);
        }
}
```

Solution

The constructor can't defined as void.

Section 9.3

9.10 How do you determine whether a character is in lowercase or uppercase?

Solution

With using `isLowerCase()` and `isUpperCase()` methods in `java.lang.Character` library.

9.11 How do you determine whether a character is alphanumeric?

Solution

With using `isLetterOrDigit()`method in `java.lang.Character` library.

Section 9.4

9.12 What is the difference between StringBuilder and StringBuffer?

Solution

There is just one difference between two of them. StringBuffer modifying buffer methods are synchronized. You should use, if you use it at more than one place.

9.13 How do you create a string builder for a string? How do you get the string from a string builder?

Solution

You can create with `StringBuilder()` or `StringBuilder( s: String )`. Using `.toString()` method returns a String from a StringBuilder.

9.14 Write three statements to reverse a string s using the reverse method in the StringBuilder class.

Solution

```
StringBuilder exampleString = StringBuilder( "I am reversed." );
exampleString.reverse();
exampleString.toString();
```

9.15 Write a statement to delete a substring from a string s of 20 characters, starting at index 4 and ending with index 10. Use the delete method in the StringBuilder class.

Solution

```
StringBuilder exampleString = StringBuilder( "I'm contain 20 chars" );
exampleString.delete( 4, 10 );
exampleString.toString();
```

9.16 What is the internal structure of a string and a string builder?

Solution

Both of them a fixed char array, because String is immutable. But in StringBuilder when string changed, Java create a new char array and assign String to it.

9.17 Suppose that s1 and s2 are given as follows: `StringBuilder s1 = new StringBuilder("Java");` `StringBuilder s2 = new StringBuilder("HTML");` Show the value of s1 after each of the following statements. Assume that the statements are independent.

(1) s1.append(" is fun");
(2) s1.append(s2);
(3) s1.insert(2, "is fun");
(4) s1.insert(1, s2);
(5) s1.charAt(2);
(6) s1.length();
(7) s1.deleteCharAt(3);
(8) s1.delete(1, 3);
(9) s1.reverse();
(10) s1.replace(1, 3, "Computer");
(11) s1.substring(1, 3);
(12) s1.substring(2);

Solution

Java is fun
JavaHTML
Jais funva
JHTMLava
v
4
Jav
Ja
avaJ
JComputera
av
va

9.18 Show the output of the following program:

```
public class Test
{
    public static void main(String[] args)
    {
        String s = "Java";
        StringBuilder builder = new StringBuilder(s);
        change(s, builder);
        System.out.println(s);
        System.out.println(builder);
    }
    private static void change(String s, StringBuilder builder)
    {
        s = s + " and HTML";
        builder.append(" and HTML");
    }
}
```

Solution

Java
Java and HTML

Section 9.5

9.19 This book declares the main method as `public static void main(String[] args)` Can it be replaced by one of the following lines?

```
public static void main(String args[])
public static void main(String[] x)
public static void main(String x[])
static void main(String x[])
```

Solution

All can but `static void main(String x[])` because it isn't public.

9.20 Show the output of the following program when invoked using

1. java Test I have a dream
2. java Test "1 2 3"
3. java Test
4. java Test "*"
5. java Test *

```java
public class Test
{
        public static void main(String[] args)
        {
                System.out.println("Number of strings is " + args.length);
                for (int i = 0; i < args.length; i++)
                System.out.println(args[i]);
        }
}
```

Solution

1. java Test I have a dream
   Number of strings is 4
   I
   have
   a
   dream

2. java Test "1 2 3"
   Number of strings is 1
   1 2 3

3. java Test
   Number of strings is 0

4. java Test "*"
   Number of strings is 1
   *

5. java Test *
   Number of strings is 2
   Test.class
   Test.java

Section 9.6

9.21 What is wrong about creating a File object using the following statement?

`new File("c:\book\test.dat");`

Solution

The backslash "\" is in Windows using as a directory separator but it is escape character in java and if you want to use string literal you should use like this: "\\"

9.22 How do you check whether a file already exists? How do you delete a file? How do you rename a file? Can you find the file size (the number of bytes) using the File class?

Solution

- You can check with if a file exist with `exampleFile.exists();` method.
- You can delete a file with `exampleFile.delete();` method.
- You can rename a file with `exampleFile.renameTo();` method.
- You can find the file size with `exampleFile.length();` method.

9.23 Can you use the File class for I/O? Does creating a File object create a file on the disk?

Solution

No. Creating a file object in Java does not create a file on the disk. With File class you can change properties and rename etc. but it can't use for I/O.

Section 9.7

9.24 How do you create a PrintWriter to write data to a file? What is the reason to declare throws Exception in the main method in Listing 9.7, WriteData.java? What would happen if the close() method were not invoked in Listing 9.7?

Solution

- You can use `new java.io.PrintWriter( fileName );` to write data to a file.
- Because if it isn't used, compiler gives error that you have to deal with this exception.
- If the `close()` method don't use data may not be saved properly.

9.25 Show the contents of the file temp.txt after the following program is executed.

```
public class Test
{
        public static void main(String[] args) throws Exception
        {
                java.io.PrintWriter output = new
                java.io.PrintWriter("temp.txt");
                output.printf("amount is %f %e\r\n", 32.32, 32.32);
                output.printf("amount is %5.4f %5.4e\r\n", 32.32, 32.32);
                output.printf("%6b\r\n", (1 > 2));
                output.printf("%6s\r\n", "Java");
                output.close();
        }
}
```

Solution

amount is 32.320000 3.232000e+01
amount is 32.3200 3.2320e+01
false
Java

9.26 How do you create a Scanner to read data from a file? What is the reason to define throws Exception in the main method in Listing 9.8, ReadData.java? What would happen if the close() method were not invoked in Listing 9.8?

Solution

- To create a Scanner to read data from file you should use `new Scanner(new File(filename));` method.
- Because if it isn't used, compiler gives error that you have to deal with this exception.
- It is not necessary to close the input file, but it is a good practice to do so to release the resources occupied by the file.

9.27 What will happen if you attempt to create a Scanner for a nonexistent file? What will happen if you attempt to create a PrintWriter for an existing file?

Solution

- Compiler gives FileNotFoundException.
- You override existing file.

9.28 Is the line separator the same on all platforms? What is the line separator on Windows?

Solution

The line-separator string is defined by the system. It is \r\n on Windows.

9.29 Suppose you enter 45 57.8 789, then press the Enter key. Show the contents of the variables after the following code is executed.

```
Scanner input = new Scanner(System.in);
int intValue = input.nextInt();
double doubleValue = input.nextDouble();
String line = input.nextLine();
```

Solution

intValue will *45*
doubleValue will *57.8*
line will *\ 789*

9.30 Suppose you enter 45, the Enter key, 57.8, the Enter key, 789, the Enter key. Show the contents of the variables after the following code is executed.

Scanner input = new Scanner(System.in);
int intValue = input.nextInt();
double doubleValue = input.nextDouble();
String line = input.nextLine();

Solution

intValue will *45*
doubleValue will *57.8*
line will