Tanmai Kalisipudi Pd 1

String Methods

For each line of code with end of line comments, record what would be stored in the declared variable or be outputted on the screen.

```
public class StringMethods
{
   public static void main(String[] args)
   {                 01234567
      String s = "Internet",  s2 = "net", s3 = "   Internet ";

      String s7 = s.substring(5);              // net
      String s8 = s.substring(0, 5);           //Inter
      String s9 = s.substring(2, 6);           // tern

      int pos11 = s.indexOf('e');              //3
      int pos12 = s.indexOf('x');              //-1
      int pos13 = s.indexOf('e', 4);           // 2
      int pos14 = s.lastIndexOf('e');          // 6
      int pos15 = s.lastIndexOf('e', 4);       // 2
      int pos16 = s.lastIndexOf('e', 2);       // 4
      int pos17 = s.indexOf(s2);               // 5
      int pos18 = s.indexOf(s2, 6);            //-1
      int pos19 = s.lastIndexOf(s2);           // 5
      int pos20 = s.lastIndexOf(s2, 6);        //-1

      boolean isSame22 = s.equals(s2);         // false
      boolean isSame23 = s.equalsIgnoreCase("internet");//true
      int result24 = s.compareTo("internet");// -32
      int result25 = s2.compareTo(s);          // 37
      int result26 = s.compareToIgnoreCase("internet");// 0

      String s28 = s.toUpperCase();            //INTERNET
      String s29 = s.replace('n', 'N');        //INterNet
      String s30 = s3.trim();                  //Internet
      // no String method changes the String object for which they are
      // called.  They build and return a new string instead. For example,
      // s3.replace('a','A') by itself is useless, because s3 remains unchanged.
      // The technical term is "immutable," as in "Strings are immutable."

      char ch = s.charAt(0);                         //I
      boolean isSame36 = (ch == 'I');                //true
      boolean isLetter37 = Character.isLetter(ch); //true
      boolean isCap38 = Character.isUpperCase(ch); //true
      char ch39 = Character.toLowerCase(ch);         // i
      String s40 = ch39 + s.substring(1);            // internet
      // three ways to visit each character of a string
      for(int i = 0; i < s.length(); i++)
         System.out.print(s.substring(i, i+1)+" ");// internet
      for(int i = 0; i < s.length(); i++)
         System.out.print(s.charAt(i)+" ");          //i n t e r n e t
      char[] chArray = s.toCharArray();
      for(int i = 0; i < chArray.length; i++)
         System.out.print(chArray[i]+" ");           // i n t e r n e t
      System.out.println();

      // Strings can be split:   String[] split(String separator)
      // The method split() returns an array of substrings split around
     // the specified separator, which is then removed
      String[] abra = "ABRACADABRA".split("A");
      for(String str : abra)
         System.out.print(str+" ");                          // BR  C DBR
      System.out.println();
      String[] abra2 = "ABRACADABRA".split("BR");
      for(String str : abra2)
         System.out.print(str+" ");                          // A  ACADA  A
      System.out.println();
      String[] abra3 = "A B R A C A D A B R A".split(" ");
      for(String str : abra3)
         System.out.println(str+" ");                        // ABRACADABRA
                                                             //

      /*   String Methods #1
      1. The string dateStr represents a date in the format "mm/dd/yyyy".
      Write a code fragment that changes dateStr to the format "dd-mm-yy".
      For example, "09/16/2008" becomes "16-09-08".
      */

                     newDate = dateStr.replace('/', '-');



      /*  String Methods #2   str
      2. Given a line of text, print each word on its own line, but don't
      print the blank lines.
      */
                     String[] x =   str.split("   ");

                     for (String y : x) {
                       System.out.println(y);

                      }
      /*   String Methods #3
      3. Given a line of text, remove all punctuation from that line.
      One way is to replace each punctuation mark with "".
      */
      String str = "RT @TJCheer2015: Freshman & Sophomores: Interested in cheer at TJ?
Email: thomasjeffersoncheer@gmail.com";
      String punct = ",./;:'\"?<>[]{}|`~!@#$%^&*()";

           String  x =  str.replaceAll(punct, "");



      /*   String Methods #4
      4. Given a line of text, remove all punctuation from that line.
      One way is to keep all characters that are letters or a space.
      */
      String str2 = "a @galaxy far, far away --right there! on the (TJ planetarium)
ceiling. https://t.co/XfoqbyA9JY";
      String letters = " ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
      String cleaned = "";  char ch;
      for(int i=0; i< str2.length ; i++) {
            ch= str2.charAt(i);
            if (Character.isLetter(ch)) {
              str2.replace("letters, "");
            }
   } }
}
```