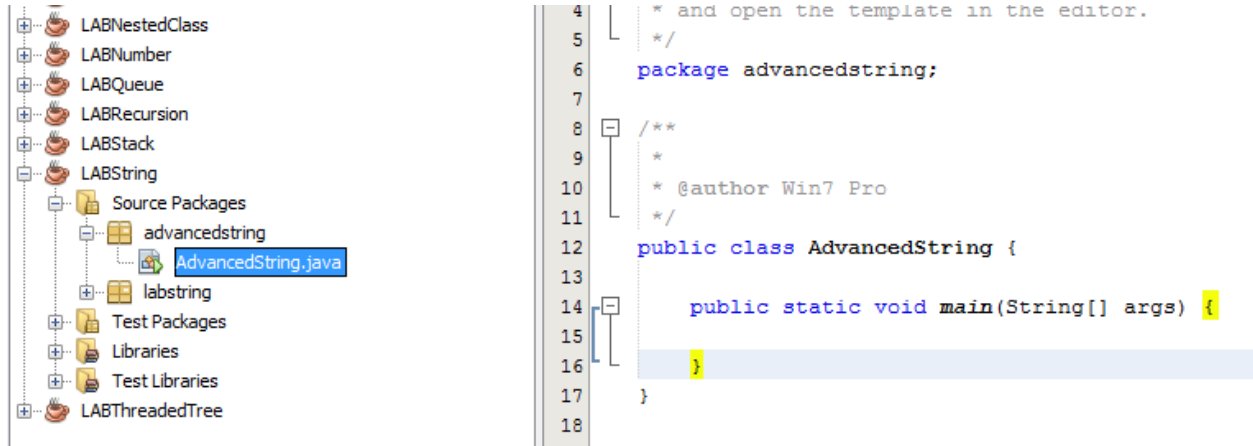


LAB

Mục tiêu: Nắm vững các hàm xử lý chuỗi trong Java

Nội dung: Thực hiện các bài tập theo yêu cầu

Xây dựng class mang tên “AdvancedString” trong đó bao gồm các phương thức tĩnh để thực hiện các câu sau



1. Xóa ký tự trắng thừa: Khi bạn nhập” Thiet Ke Trang Chu .Com” kết quả sẽ là ThietKeTrangChu.Com

```
public static String cau1(String s) {
    StringTokenizer strToken = new StringTokenizer(s, " ");

    String result = "";

    while(strToken.hasMoreTokens()) {
        result += strToken.nextToken();
    }

    return result;
}
```

Kiểm tra với hàm main

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

    System.out.println(AdvancedString.cau1("Hello FPT University"));
}
```

Email: giangdayit@gmail.com

FB: <https://www.facebook.com/giangdaytinhoc>

2. Kiểm tra xem chuỗi có đối xứng hay không

```
public static boolean cau2(String s) {  
    boolean result = s.equals(new StringBuffer(s).reverse().toString());  
    return result;  
}
```

3. Viết 1 chương trình đếm một kí tự xuất hiện bao nhiêu lần trong chuỗi.
4. Nhập một chuỗi bất kì, yêu cầu nhập 1 kí tự muốn xóa. Thực hiện xóa tất cả những kí tự đó trong chuỗi.
5. Cho một xâu, nhập vào một từ, viết chương trình, xóa từ đó trong xâu.
6. Chương trình tìm kiếm xem ký tự nào xuất hiện nhiều nhất trong chuỗi.
7. Nhập 1 chuỗi bất kì, liệt kê xem mỗi ký tự xuất hiện mấy lần
8. Nhập vào chuỗi s1 và s2, cho biết vị trí xuất hiện của chuỗi s2 trong s1.

----- ADVANCED LAB -----

9. Write an application that reads a line of text from the keyboard and prints a table indicating the number of occurrences of each letter of the alphabet in the text. For example, the phrase **To be, or not to be: that is the question :** contains one "a," two "b's," no "c's," and so on.
10. Write an application that reads a line of text and prints a table indicating the number of one-letter words, two-letter words, three-letter words, and so on, appearing in the text. For example, Fig. 29.25 shows the counts for the phrase. Figure 29.25. Word-length counts for the string **"Whether 'tis nobler in the mind to suffer"**.

Email: giangdayit@gmail.com

FB: <https://www.facebook.com/giangdaytinhoc>

Word length	Occurrences
1	0
2	2
3	1
4	2 (including 'tis)
5	0
6	2
7	1

11. Write an application that reads a line of text and prints a table indicating the number of occurrences of each different word in the text. The first version of your application should include the words in the table in the same order in which they appear in the text. For example, the lines **To be, or not to be: that is the question:** Whether 'tis nobler in the mind to suffer contain the word "to" three times, the word "be" two times, the word "or" once, and so on. A more interesting (and useful) printout should then be attempted in which the words are sorted alphabetically.