ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

OBJECT-ORIENTED PROGRAMMING

StringBuffer – StringBuilder - StringTokenizer

Outline

- StringBuffer
- StringBuilder
- String and StringBuffer
- StringBuffer and StringBuilder
- StringTokenizer

StringBuffer

- Java StringBuffer class is used to create mutable (modifiable) string. The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.
- Java StringBuffer class is thread-safe i.e. multiple threads cannot access it simultaneously. So it is safe and will result in an order

StringBuffer

- Constructor:

Constructor	Description
StringBuffer()	creates an empty string buffer with the initial capacity of 16.
StringBuffer(String str)	creates a string buffer with the specified string.
StringBuffer(int capacity)	creates an empty string buffer with the specified capacity as length

StringBuffer

- Important methods

Modifier and Type	Method	Description
public synchronized StringBuffer	append(String s)	is used to append the specified string with this string. The append() method is overloaded like append(char), append(boolean), append(int), append(float), append(double) etc.
public synchronized StringBuffer	insert(int offset, String s)	is used to insert the specified string with this string at the specified position. The insert() method is overloaded like insert(int, char), insert(int, boolean), insert(int, int), insert(int, float), insert(int, double) etc.
public synchronized StringBuffer	replace(int startIndex, int endIndex, String str)	is used to replace the string from specified startIndex and endIndex.
public synchronized StringBuffer	delete(int startIndex, int endIndex)	is used to delete the string from specified startIndex and endIndex.
public synchronized StringBuffer	reverse()	is used to reverse the string.

StringBuffer

- Important methods

public int	capacity()	is used to return the current capacity.
public void	ensureCapacity(int minimumCapacity)	is used to ensure the capacity at least equal to the given minimum.
public char	charAt(int index)	is used to return the character at the specified position.
public int	length()	is used to return the length of the string i.e. total number of characters.
public String	substring(int beginIndex)	is used to return the substring from the specified beginIndex.
public String	substring(int beginIndex, int endIndex)	is used to return the substring from the specified beginIndex and endIndex.

StringBuffer

- append()

```
class StringBufferExample{
        public static void main(String args[]){
                StringBuffer sb=new StringBuffer("Hello ");

        sb.append("Java");//now original string is changed
                System.out.println(sb);//prints Hello Java
        }
}
```

StringBuffer

- insert()

```
class StringBufferExample2{
    public static void main(String args[]){
        StringBuffer sb=new StringBuffer("Hello ");
        sb.insert(1,"Java");//now original string is changed
        System.out.println(sb);//prints HJavaello
    }
}
```

StringBuffer

- replace()

```
class StringBufferExample3{
    public static void main(String args[]){
        StringBuffer sb=new StringBuffer("Hello");
        sb.replace(1,3,"Java");
        System.out.println(sb);//prints HJavalo
    }
}
```

StringBuffer

- delete()

```
class StringBufferExample4{
    public static void main(String args[]){
        StringBuffer sb=new StringBuffer("Hello");
        sb.delete(1,3);
        System.out.println(sb);//prints Hlo
    }
}
```

StringBuffer

- reverse()

```
class StringBufferExample5{
    public static void main(String args[]){
        StringBuffer sb=new StringBuffer("Hello");
        sb.reverse();
        System.out.println(sb);//prints olleH
    }
}
```

StringBuilder

- Java StringBuilder class is used to create mutable (modifiable) string. The Java StringBuilder class is same as StringBuffer class except that it is non-synchronized. It is available since JDK 1.5.

Constructor	Description
StringBuilder()	creates an empty string Builder with the initial capacity of 16.
StringBuilder(String str)	creates a string Builder with the specified string.
StringBuilder(int length)	creates an empty string Builder with the specified capacity as length.

StringBuilder

Method	Description
public StringBuilder append(String s)	is used to append the specified string with this string. The append() method is overloaded like append(char), append(boolean), append(int), append(float), append(double) etc.
public StringBuilder insert(int offset, String s)	is used to insert the specified string with this string at the specified position. The insert() method is overloaded like insert(int, char), insert(int, boolean), insert(int, int), insert(int, float), insert(int, double) etc.
public StringBuilder replace(int startIndex, int endIndex, String str)	is used to replace the string from specified startIndex and endIndex.
public StringBuilder delete(int startIndex, int endIndex)	is used to delete the string from specified startIndex and endIndex.
public StringBuilder reverse()	is used to reverse the string.
public int capacity()	is used to return the current capacity.
public void ensureCapacity(int minimumCapacity)	is used to ensure the capacity at least equal to the given minimum.
public char charAt(int index)	is used to return the character at the specified position.

StringBuilder

public int length()	is used to return the length of the string i.e. total number of characters.
public String substring(int beginIndex)	is used to return the substring from the specified beginIndex.
public String substring(int beginIndex, int endIndex)	is used to return the substring from the specified beginIndex and endIndex.

- The examples of StringBuilder class are similar to StringBuffer's.

Exercises

Use StringBuffer, StringBuilder and StringTokenizer to complete these requirement

Give a Vietnamese fullname (ex “Nguyen Van Teo”).
Students need to write methods as follows:

1. Count how many words in the name.
2. Return a first name
3. Return a last name
4. Return a middle name
5. Capitalize the first character in each word of the name
6. Formailize the name, including:
   - Delete spaces in front and behind of the name.
   - Leave one space between the words of the name.

String and StringBuffer

- There are many differences between String and StringBuffer. A list of differences between String and StringBuffer are given below:

No.	String	StringBuffer
1)	String class is immutable.	StringBuffer class is mutable.
2)	String is slow and consumes more memory when you concat too many strings because every time it creates new instance.	StringBuffer is fast and consumes less memory when you cancat strings.
3)	String class overrides the equals() method of Object class. So you can compare the contents of two strings by equals() method.	StringBuffer class doesn't override the equals() method of Object class.

StringBuffer and String Builder

- Java provides three classes to represent a sequence of characters: String, StringBuffer, and StringBuilder. The String class is an immutable class whereas StringBuffer and StringBuilder classes are mutable. There are many differences between StringBuffer and StringBuilder. The StringBuilder class is introduced since JDK 1.5.
- A list of differences between StringBuffer and StringBuilder are given below:

No.	StringBuffer	StringBuilder
1)	StringBuffer is *synchronized* i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously.	StringBuilder is *non-synchronized* i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously.
2)	StringBuffer is *less efficient* than StringBuilder.	StringBuilder is *more efficient* than StringBuffer.

StringTokenizer

- The **java.util.StringTokenizer** class allows you to break a string into tokens. It is simple way to break string.

Constructor	Description
StringTokenizer(String str)	creates StringTokenizer with specified string.
StringTokenizer(String str, String delim)	creates StringTokenizer with specified string and delimeter.
StringTokenizer(String str, String delim, boolean returnValue)	creates StringTokenizer with specified string, delimeter and returnValue. If return value is true, delimiter characters are considered to be tokens. If it is false, delimiter characters serve to separate tokens.

StringTokenizer

Public method	Description
boolean hasMoreTokens()	checks if there is more tokens available.
String nextToken()	returns the next token from the StringTokenizer object.
String nextToken(String delim)	returns the next token based on the delimeter.
boolean hasMoreElements()	same as hasMoreTokens() method.
Object nextElement()	same as nextToken() but its return type is Object.
int countTokens()	returns the total number of tokens.

StringTokenizer

```
import java.util.StringTokenizer;
public class Simple{
 public static void main(String args[]){
   StringTokenizer st = new StringTokenizer("my name is kh
an"," ");
     while (st.hasMoreTokens()) {
         System.out.println(st.nextToken());
     }
   }
}
```

```
Output:my
       name
       is
       khan
```

StringTokenizer

```
import java.util.*;
public class Test {
   public static void main(String[] args) {
       StringTokenizer st = new StringTokenizer("my,name,is,khan");

      // printing next token
      System.out.println("Next token is : " + st.nextToken(","));
   }
}
```

```
Output:Next token is : my
```

Exercises

Give a Vietnamese fullname (ex “Nguyen Van Teo”).

Using StringBuffer or StringBuilder or StringTokenizer to complete these exercises, Students need to write methods as follows:

1. Count how many words in the name.
2. Return a first name
3. Return a last name
4. Return a middle name
5. Capitalize the first character in each word of the name
6. Formailize the name, including:
   - Delete spaces in front and behind of the name.
   - Leave one space between the words of the name.

End of file