

+ Java Cơ Bản

+ Các Khái Niệm Java OOPs

+ Java String

- String là gì
- Immutable String
- So sánh String
- Nội String
- Sub-String
- Phương thức của lớp String
- Lớp StringBuffer**
- Lớp StringBuilder
- String vs StringBuffer
- StringBuffer vs Builder
- Tạo lớp Immutable
- toString method
- Lớp StringTokenizer

+ Phương thức Java String

- String charAt()
- String compareTo()
- String concat()
- String contains()
- String endsWith()
- String equals()
- equalsIgnoreCase()
- String format()
- String getBytes()
- String getChars()
- String indexOf()
- String intern()
- String isEmpty()
- String join()
- String lastIndexOf()
- String length()
- String replace()
- String replaceAll()
- String split()
- String startsWith()
- String substring()
- String toCharArray()
- String toLowerCase()
- String toUpperCase()
- String trim()
- String valueOf()

+ Xử Lý Ngoại Lệ

Lớp StringBuffer trong java

☺ Các phương thức của lớp String trong Java

Lớp StringBuilder trong java ☺

Nội dung chính:

Lớp StringBuffer trong java

- Các Constructor quan trọng của lớp StringBuffer trong java
- Các phương thức của lớp StringBuffer trong java
- Chuỗi có thể sửa đổi là gì?
- 1) Phương thức append() của lớp StringBuffer trong java
- 2) Phương thức insert() của lớp StringBuffer trong java
- 3) Phương thức replace() của lớp StringBuffer trong java
- 4) Phương thức delete() của lớp StringBuffer trong java
- 5) Phương thức reverse() của lớp StringBuffer trong java
- 6) Phương thức capacity() của lớp StringBuffer trong java
- 7) Phương thức ensureCapacity() của lớp StringBuffer trong java

Lớp StringBuffer trong java

Trong java, lớp StringBuffer được sử dụng để tạo chuỗi có thể thay đổi (mutable). Lớp StringBuffer trong java tương tự như lớp String ngoại trừ nó có thể thay đổi.



Chú ý: Lớp StringBuffer là thread-safe (luồng an toàn) nghĩa là nhiều luồng (thread) không thể truy cập nó trong cùng một thời điểm.

Các Constructor quan trọng của lớp StringBuffer trong java

- StringBuffer():** Tạo ra một bộ đệm chuỗi với dung lượng ban đầu là 16.
- StringBuffer(String str):** Tạo ra một bộ đệm chuỗi với chuỗi cụ thể.
- StringBuffer(int capacity):** Tạo ra một bộ đệm chuỗi với dung lượng được chỉ định như độ dài chuỗi.

Các phương thức của lớp StringBuffer trong java

- public synchronized StringBuffer append(String s):** được sử dụng để nối thêm các chuỗi được chỉ định với chuỗi này. Các phương thức append() được nạp chồng như append(char), append(boolean), append(int), append(float), append(double), ...
- public synchronized StringBuffer insert(int offset, String s):** được sử dụng để chèn chuỗi chỉ định với chuỗi này tại vị trí quy định. Các phương thức insert() được nạp chồng như insert(int, char), insert(int, boolean), insert(int, int), insert(int, float), insert(int, double), ...
- public synchronized StringBuffer replace(int startIndex, int endIndex, String str):** được sử dụng để

Recent Updates

- ☺ Đọc file CSV trong Python
- ☺ Lệnh DELETE MySQL trong Python
- ☺ Lệnh UPDATE MySQL trong Python
- ☺ Lệnh SELECT MySQL trong Python
- ☺ Lệnh INSERT MySQL trong Python
- ☺ Cài đặt môi trường MySQL cho Python
- ☺ Tạo bảng MySQL trong Python
- ☺ Tạo new database MySQL trong Python
- ☺ Kết nối Python với MySQL
- ☺ Trừu tượng dữ liệu trong Python
- ☺ Ghi đè phương thức trong Python - Method Overriding
- ☺ Kế thừa trong Python

VietTuts on facebook



Bạn và 1 người bạn khác thích nội dung này



thay thế chuỗi từ vị trí startIndex đến endIndex bằng chuỗi str.

4. **public synchronized StringBuffer delete(int startIndex, int endIndex):** được sử dụng để xóa chuỗi từ vị trí startIndex đến endIndex.
5. **public synchronized StringBuffer reverse():** được sử dụng để đảo ngược chuỗi.
6. **public int capacity():** được sử dụng để trả về dung lượng hiện tại.
7. **public void ensureCapacity(int minimumCapacity):** được sử dụng để đảm bảo dung lượng ít nhất bằng mức tối thiểu nhất định.
8. **public char charAt(int index):** được sử dụng trả về ký tự tại vị trí quy định.
9. **public int length():** được sử dụng trả về chiều dài của chuỗi nghĩa là tổng số ký tự.
10. **public String substring(int beginIndex):** được sử dụng trả về chuỗi con bắt đầu từ vị trí được chỉ định.
11. **public String substring(int beginIndex, int endIndex):** được sử dụng trả về chuỗi con với vị trí bắt đầu và vị trí kết thúc được chỉ định.

Chuỗi có thể sửa đổi là gì?

Một chuỗi có thể chỉnh sửa hoặc thay đổi được gọi là chuỗi có thể sửa đổi. Các lớp StringBuffer và StringBuilder được sử dụng để tạo ra chuỗi có thể sửa đổi(mutable).

1) Phương thức append() của lớp StringBuffer trong java

Phương thức append() của lớp StringBuffer nối thêm tham số vào cuối chuỗi.

```
public class StringBufferExam1 {
    public static void main(String args[]) {
        StringBuffer sb = new StringBuffer("Hello ");
        sb.append("Java");//đến đây chuỗi ban đầu đã bị thay đổi
        System.out.println(sb);//in Hello Java
    }
}
```

2) Phương thức insert() của lớp StringBuffer trong java

Phương thức insert() của lớp StringBuffer chèn chuỗi vào chuỗi này từ vị trí quy định.

```
public class StringBufferExam2 {
    public static void main(String args[]) {
        StringBuffer sb = new StringBuffer("Hello ");
        sb.insert(1, "Java");//đến đây chuỗi ban đầu đã bị thay đổi
        System.out.println(sb);//in -> HJavaello
    }
}
```

3) Phương thức replace() của lớp StringBuffer trong java

Phương thức replace() của lớp StringBuffer thay thế chuỗi bằng chuỗi khác từ vị trí bắt đầu và kết thúc được quy định.

```
public class StringBufferExam3 {
    public static void main(String args[]) {
        StringBuffer sb = new StringBuffer("Hello");
        sb.replace(1, 3, "Java");
    }
}
```

```

        sb.replace(1, 3, "Java");
        System.out.println(sb);//in -> HJavaLo
    }
}

```

4) Phương thức delete() của lớp StringBuffer trong java

Phương thức replace() của lớp StringBuffer xóa chuỗi từ vị trí bắt đầu và kết thúc được quy định.

```

public class StringBufferExam4 {
    public static void main(String args[]) {
        StringBuffer sb = new StringBuffer("Hello");
        sb.delete(1, 3);
        System.out.println(sb);//in -> Hlo
    }
}

```

5) Phương thức reverse() của lớp StringBuffer trong java

Phương thức reverse() của lớp StringBuffer đảo ngược chuỗi hiện tại.

```

public class StringBufferExam5 {
    public static void main(String args[]) {
        StringBuffer sb = new StringBuffer("Hello");
        sb.reverse();
        System.out.println(sb);//in -> olleH
    }
}

```

6) Phương thức capacity() của lớp StringBuffer trong java

Phương thức capacity() của lớp StringBuffer trả về dung lượng của bộ nhớ đệm. Dung lượng mặc định của bộ nhớ đệm là 16. Nếu số lượng ký tự của chuỗi tăng lên thì dung lượng được tính theo công thức (dung lượng cũ*2)+2. Ví dụ: Nếu dung lượng hiện tại là 16, nó sẽ tăng lên (16*2)+2=34.

```

public class StringBufferExam6 {
    public static void main(String args[]) {
        StringBuffer sb = new StringBuffer();
        System.out.println(sb.capacity());//mặc định là 16
        sb.append("Hello");
        System.out.println(sb.capacity());//đến đây vẫn là 16
        sb.append("java is my favourite language");
        System.out.println(sb.capacity());//đến đây là (16*2)+2=34 i.e (dung lượng cũ
    }
}

```

7) Phương thức ensureCapacity() của lớp StringBuffer trong java

Phương thức ensureCapacity() của lớp StringBuffer đảm bảo rằng dung lượng đã cho là tối thiểu với dung lượng hiện tại. Nếu nó lớn hơn dung lượng hiện tại, dung lượng hiện tại được tăng theo công thức (dung lượng cũ*2)+2. Ví dụ, dung lượng hiện tại là 16, nó sẽ tăng lên là (16*2)+2=34.

```

public class StringBufferExam7 {
    public static void main(String args[]) {

```

```
public static void main(String args[]) {
    StringBuffer sb = new StringBuffer();
    System.out.println(sb.capacity()); //mặc định là 16
    sb.append("Hello");
    System.out.println(sb.capacity()); //đến đây là 16
    sb.append("java is my favourite language");
    System.out.println(sb.capacity()); //đến đây là (16*2)+2=34 i.e (dung lượng cũ
    sb.ensureCapacity(10); //đến đây không có sự thay đổi
    System.out.println(sb.capacity()); //đến đây là 34
    sb.ensureCapacity(50); //đến đây là (34*2)+2
    System.out.println(sb.capacity()); //đến đây là 70
}
```

Tìm hiểu về xử lý chuỗi trong java

- String là gì
- Immutable String
- So sánh String
- Nối String
- Sub-String
- Phương thức của lớp String
- Lớp StringBuffer
- Lớp StringBuilder
- String vs StringBuffer
- StringBuffer vs Builder
- Tạo lớp Immutable
- Phương thức toString
- Lớp StringTokenizer

⊕ Các phương thức của lớp String trong Java

```
equals()  
&  
hashCode()
```



Phương thức equals() và hashCode() trong java

Swing trong Java

thiệu Java Swing



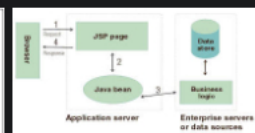
ArrayList trong java - học Java miễn phí hay nhất



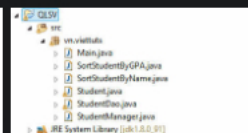
Tuyển Dụng Công Nghệ Thông Tin



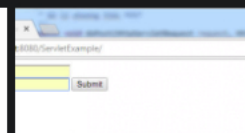
Servlet là gì? - tại sao bạn nên biết servlet



JSP là gì? - học JS chuẩn bị những gì



**Bài tập quản lý sinh viên
trong java - bài tập java
có lời giải**



n Ví dụ phương thức POST sử dụng Form - học servlet cơ bản...



0 bình luận

Sắp xếp theo

Cũ nhất



Thêm bình luận...



Plugin bình luận trên Facebook

Danh sách bài học

Học java
Học servlet
Học jsp
Học Hibernate
Học Struts2
Học Spring
Học SQL

Câu hỏi phỏng vấn

201 câu hỏi phỏng vấn java
25 câu hỏi phỏng vấn servlet
75 câu hỏi phỏng vấn jsp
52 câu hỏi phỏng vấn Hibernate
70 câu hỏi phỏng vấn Spring
57 câu hỏi phỏng vấn SQL

About VietTuts.Vn

Hệ thống bài học trên VietTuts.Vn bao gồm các bài lý thuyết và thực hành về các công nghệ java và công nghệ web. Các bài lý thuyết trên hệ thống VietTuts.Vn được tham khảo và tổng hợp từ các trang <http://javatpoint.com>, <http://www.tutorialspoint.com>, <http://docs.oracle.com/en> ...