



BÀI GIẢNG

NGÔN NGỮ LẬP TRÌNH JAVA

CHƯƠNG 5:

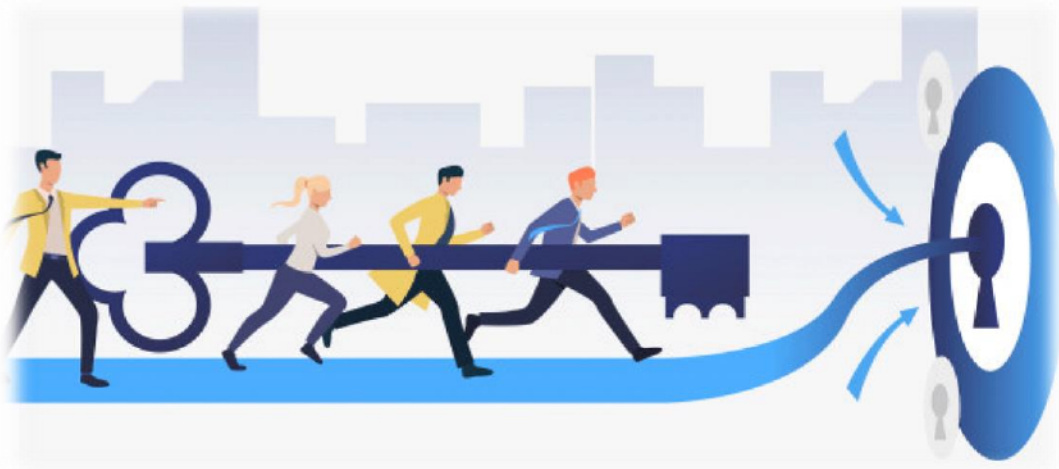
CHUỖI VÀ BIỂU THỨC CHÍNH QUY

*TS. Nguyễn Sĩ Thìn
(nsthin@vku.udn.vn)*

Chương 5. Chuỗi và biểu thức chính quy

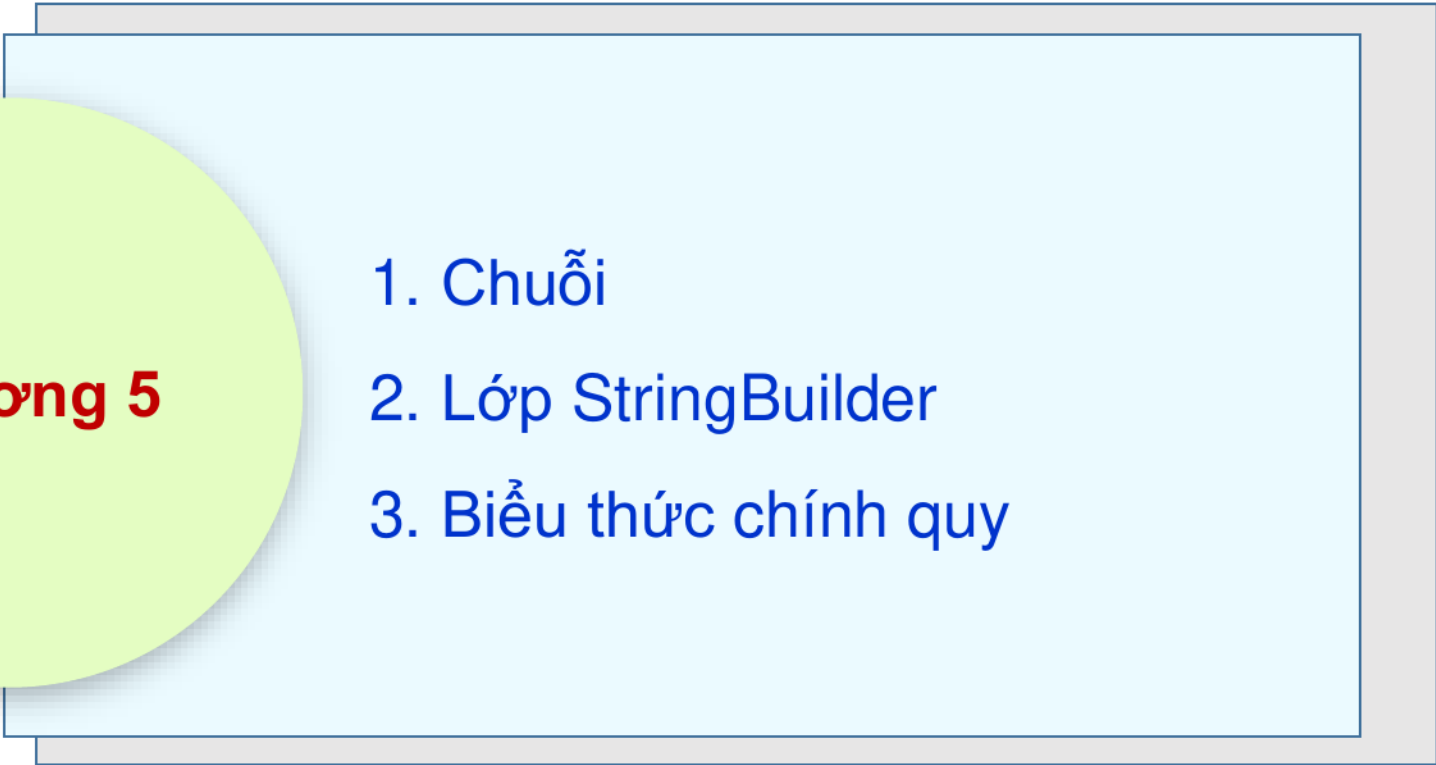
Mục tiêu

- 🎯 Hiểu và sử dụng được chuỗi
- 🎯 Hiểu và sử dụng được lớp StringBuilder
- 🎯 Hiểu và sử dụng được biểu thức chính quy

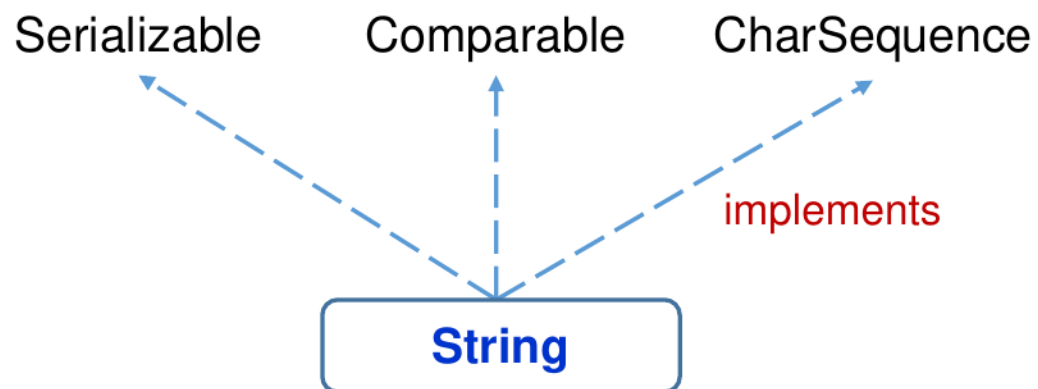




Chương 5

- 
1. Chuỗi
 2. Lớp StringBuilder
 3. Biểu thức chính quy

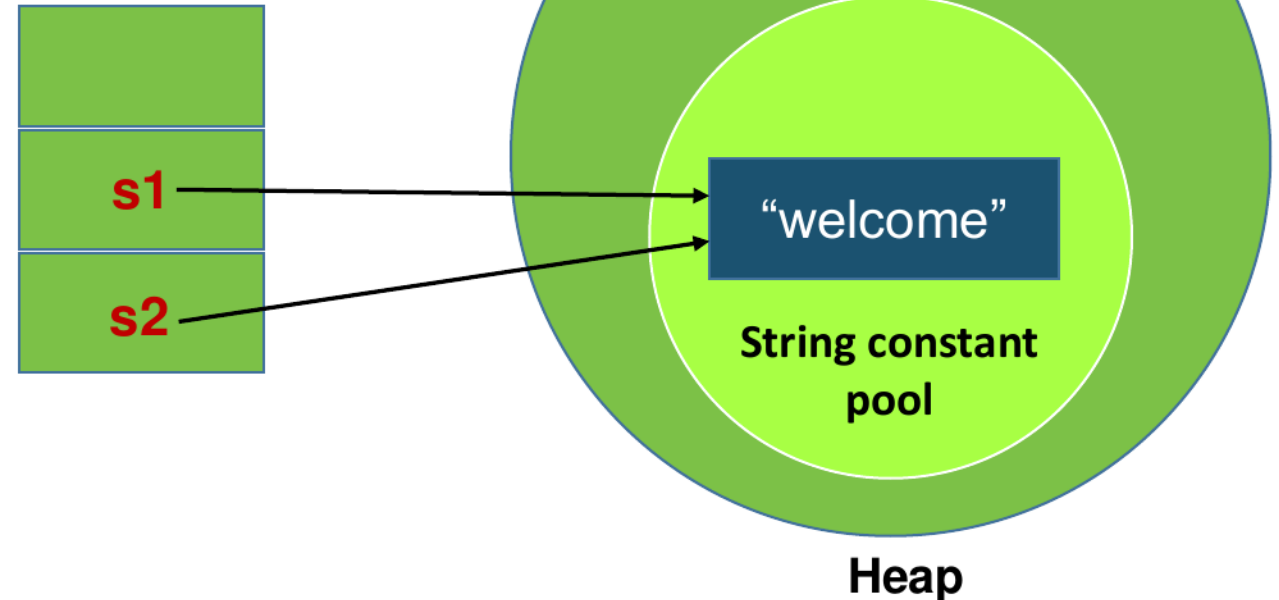
- **Khái niệm:** Là một đối tượng lưu giữ một xâu ký tự
- **Lưu ý:**
 - Không phải mảng các ký tự
 - `java.lang.String`
 - Lớp String cung cấp nhiều phương thức xử lý xâu ký tự
 - Bất biến (immutable)



1.1 Khai báo chuỗi

- **Cách 1:** Dùng hằng chuỗi
- **Cú pháp:** String tenChuoi = “Giá trị khởi tạo”;
- **Lưu ý:** Hằng chuỗi được lưu trong “String constant pool”
- **Ví dụ:**

```
String s1="Welcome";  
String s2="Welcome";  
//không tạo ra một đối tượng mới
```



- **Cách 2:** Dùng toán tử “new”
- **Cú pháp:** `String tenChuoi=new String(“Giá trị khởi tạo”);`
- **Ví dụ 1:**

```
String s=new String("Welcome");  
//tạo ra 2 đối tượng String và một biến tham chiếu s
```

- **Ví dụ 2:**

```
String s1="java";           // tạo ra một chuỗi với hằng chuỗi  
char ch[]={'s','t','r','i','n','g','s'};  
String s2=new String(ch);   //chuyển mảng char sang chuỗi  
String s3=new String("example"); //tạo chuỗi bằng từ khóa new  
System.out.println(s1);     // java  
System.out.println(s2);     // strings  
System.out.println(s3);     // example
```



1.2 Các phương thức của lớp String

Các phương thức so sánh của lớp String:

- **Phương thức equals():**
 - So sánh giá trị của các đối tượng chuỗi
 - Có 2 phương thức:
 - `public boolean equals(Object another)`
 - `public boolean equalsIgnoreCase(String another)`
- **Toán tử “==”:**
 - So sánh tham chiếu, không phải giá trị
- **Phương thức compareTo():**
 - So sánh giá trị giữa 2 chuỗi s1 và s2
 - Giá trị trả về:
 - Số “0” nếu s1==s2
 - Số dương nếu s1>s2
 - Số âm nếu s1<s2



Phương thức nối chuỗi lớp String

▪ Sử dụng toán tử “+”:

- Có thể nối với các giá trị kiểu nguyên thủy
- Sau một hằng chuỗi các toán hạng “+” được xem như là phép nối chuỗi

○ Ví dụ:

```
String s=50+30+"Sachin"+40+40;  
System.out.println(s);  
//Kết quả màn hình: 80Sachin4040
```

▪ Phương thức “concat()”:

- Nối chuỗi này vào cuối chuỗi kia

○ Ví dụ:

```
String s1="Hello ";  
String s2="World";  
String s3=s1.concat(s2);  
System.out.println(s3);  
//Kết quả: Hello World
```

▪ Phương thức “String.format()”

▪ Phương thức “String.join()”



Phương thức lấy một chuỗi con

- **Chuỗi con (substring):** Là một phần (tập con) của chuỗi ban đầu
- **Phương thức “substring()”:**
 - `public String substring(int startIndex)`
→ Lấy ra chuỗi con bắt đầu từ vị trí “startIndex”
 - `public String substring(int startIndex, int endIndex)`
→ Lấy ra chuỗi con từ vị trí “startIndex” đến vị trí “endIndex - 1”
- **Phương thức “String.split()”**



Một số phương thức cơ bản khác của lớp String

TT	Phương thức	Ý nghĩa
1	<code>char charAt(int index)</code>	Trả về kí tự từ vị trí đã cho
2	<code>int length()</code>	Trả về độ dài chuỗi
3	<code>boolean contains(CharSequence s)</code>	Kiểm tra có hay không một chuỗi con trong chuỗi
4	<code>boolean isEmpty()</code>	Kiểm tra có phải chuỗi có độ dài bằng 0
5	<code>String replace(char old, char new)</code>	Thay thế tất cả ký tự “old” bằng ký tự “new”
6	<code>String replace(CharSequence old, CharSequence new)</code>	Thay thế tất cả chuỗi con cũ “old” bằng chuỗi con mới “new”
7	<code>String trim()</code>	Xóa đi các ký tự trắng thừa của chuỗi
8	<code>static String valueOf(int value)</code>	Chuyển kiểu số nguyên “int” sang kiểu chuỗi “String”

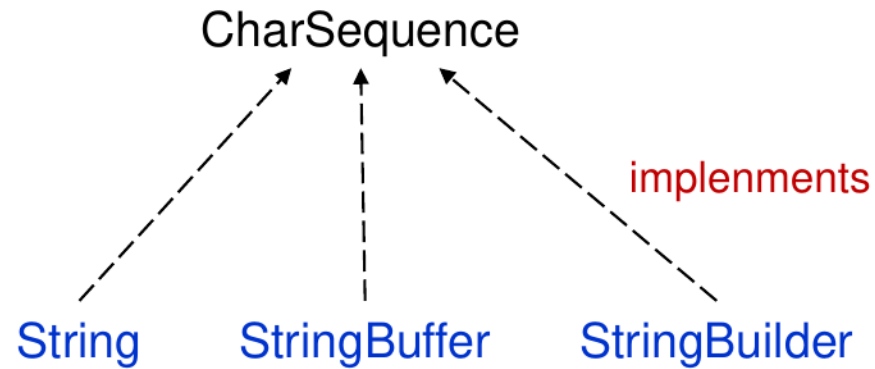


Một số phương thức cơ bản khác của lớp String (tt)

TT	Phương thức	Ý nghĩa
9	<code>int indexOf(int ch)</code>	Trả về vị trí của ký tự đã cho
10	<code>int indexOf(int ch, int fromIndex)</code>	Trả về vị trí của ký tự đã cho tính từ vị trí chỉ bắt đầu chỉ định
11	<code>int indexOf(String substring)</code>	Trả về vị trí của chuỗi đã cho
12	<code>int indexOf(String substring, int fromIndex)</code>	Trả về vị trí của chuỗi đã cho tính từ vị trí bắt đầu chỉ định
13	<code>String toLowerCase()</code>	Chuyển thành kiểu viết thường tất cả các ký tự viết hoa
14	<code>String toLowerCase(Locale l)</code>	Chuyển thành kiểu viết thường tại phạm vi được chỉ định
15	<code>String toUpperCase()</code>	Chuyển thành kiểu viết hoa tất cả các ký tự viết thường
16	<code>String toUpperCase(Locale l)</code>	Chuyển thành kiểu viết hoa tại phạm vi được chỉ định



2. Lớp StringBuilder



▪ Lớp **StringBuilder**:

- Tạo ra các chuỗi có thể chỉnh sửa được
- Cơ bản tương tự lớp `StringBuffer` nhưng không có tính đồng bộ
- So sánh với lớp `String`, ở **`StringBuider`** thì:
 - Không có phương thức “`equals()`”
 - Không sử dụng vùng nhớ “String constant pool”

Hàm dựng của lớp StringBuilder

▪ Hàm dựng:

- `StringBuilder();`
→ Tạo đối tượng mới trống với khả năng chứa mặc định là 16
- `StringBuilder(String str);`
→ Tạo đối tượng mới với giá trị chuỗi được cho trước
- `StringBuilder(int length);`
→ Tạo đối tượng mới trống với khả năng chứa cho trước



Một số phương thức cơ bản của lớp StringBuilder

STT	Phương thức	Ý nghĩa
1	<code>public StringBuilder append(String s)</code>	Nối một chuỗi vào vị trí cuối của chuỗi ban đầu
2	<code>public StringBuilder insert(int offset, String s)</code>	Chèn một chuỗi vào chuỗi ban đầu tại vị trí chỉ định
4	<code>public StringBuilder delete(int startIndex, int endIndex)</code>	Xóa chuỗi từ vị trí bắt đầu và kết thúc chỉ định
5	<code>public StringBuilder reverse()</code>	Đảo ngược chuỗi



3. Biểu thức chính quy

- **Biểu thức chính quy (Java Regex / Regular Expression):**
 - Là một chuỗi mẫu dùng để qui định dạng thức của các chuỗi
 - Gồm:
 - 01 interface: MatcherResult
 - Gói “*java.util.regex*” chứa 3 class:
 - + Matcher
 - + Pattern
 - + PatternSyntaxException



- **Đề bài:** Anh/Chị hãy viết chương trình yêu cầu người dùng nhập thông tin số điện thoại, email, ngày tháng năm sinh và kiểm tra yêu cầu định dạng.
- **Gợi ý:**
 - Sử dụng Pattern và quy định định dạng như sau:
 - + SĐT: 0xxx-xxx-xxx
 - + Email: abc@xxx.xxx.xx;
 - + Ngày tháng năm sinh: dd/MM/yyyy

THANK
YOU!
