



LẬP TRÌNH JAVA 1

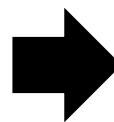
BÀI 7: CHUỖI VÀ BIỂU THỨC CHÍNH QUI PHẦN 1

- ⊙ Hiểu và sử dụng chuỗi
- ⊙ Hiểu và sử dụng biểu thức chính qui

- ❑ String là xâu các ký tự.
 - ❖ String s = "Hello World";
- ❑ String là một class được xây dựng sẵn trong Java. String có rất nhiều phương thức giúp xử lý chuỗi một cách thuận tiện và hiệu quả.
- ❑ String là kiểu dữ liệu được sử dụng nhiều nhất trong lập trình

| Ký tự | Hiển thị |
|-------|-------------|
| \t | Ký tự tab |
| \r | Về đầu dòng |
| \n | Xuống dòng |
| \\ | \ |
| \" | " |

```
System.out.print("\t+ Họ và tên: Tuấn\r\n\t+ Tuổi: 40");
```



```
+ Họ và tên: Tuấn  
+ Tuổi: 40
```

- ❑ So sánh
- ❑ Tìm vị trí của chuỗi con
- ❑ Lấy chuỗi con
- ❑ Tách và hợp chuỗi
- ❑ Chuyển đổi hoa thường
- ❑ Lấy độ dài
- ❑ ...

```
String fullname = "Nguyễn Văn Tèo";  
String first = fullname.substring(0, 6);
```

↓
Nguyễn

| Phương thức | Mô tả |
|---------------------------|---------------------------------|
| toLowerCase () | Đổi in thường |
| toUpperCase () | Đổi in hoa |
| trim() | Cắt các ký tự trắng 2 đầu chuỗi |
| length() | Lấy độ dài chuỗi |
| substring() | Lấy chuỗi con |
| charAt (index) | Lấy ký tự tại vị trí |
| replaceAll(find, replace) | Tìm kiếm và thay thế tất cả |
| split(separator) | Tách chuỗi thành mảng |

| Phương thức | Mô tả |
|---------------------------------|--|
| <code>equals()</code> | So sánh bằng có phân biệt hoa/thường |
| <code>equalsIgnoreCase()</code> | So sánh bằng không phân biệt hoa/thường |
| <code>contains()</code> | Kiểm tra có chứa hay không |
| <code>startsWith()</code> | Kiểm tra có bắt đầu bởi hay không |
| <code>endsWith ()</code> | Kiểm tra có kết thúc bởi hay không |
| <code>matches ()</code> | So khớp với hay không? |
| <code>indexOf()</code> | Tìm vị trí xuất hiện đầu tiên của chuỗi con |
| <code>lastIndexOf()</code> | Tìm vị trí xuất hiện cuối cùng của chuỗi con |

- ❑ Đăng nhập hợp lệ khi mã tài khoản là "hello" và mật khẩu trên 6 ký tự
- ❑ Thực hiện:
 - ❖ Nhập username và password từ bàn phím
 - ❖ Sử dụng `equalsIgnoreCase()` để so sánh username và `length()` để lấy độ dài mật khẩu

```
if(username.equalsIgnoreCase("hello") && password.length() > 6){  
    ...  
}  
else{  
    ...  
}
```


❑ Quản lý sinh viên

- ❖ Nhập mảng họ tên sinh viên
- ❖ Xuất họ và tên (IN HOA) những sinh viên tên Tuấn hoặc họ Nguyễn
- ❖ Xuất tên những sinh viên có tên lót là Mỹ

❑ Thự hiện

- ❖ `fullname.toUpperCase()`: đổi IN HOA
- ❖ `fullname.startsWith("Nguyễn ")`: họ Nguyễn
- ❖ `fullname.endsWith(" Tuấn")`: tên Tuấn
- ❖ `fullname.contains(" Mỹ ")`: lót Mỹ
- ❖ `fullname.lastIndexOf(" ")`: Lấy vị trí trắng cuối cùng
- ❖ `fullname.substring(lastIndex + 1)`: Lấy tên

- ❑ Tìm kiếm và thay thế chuỗi
- ❑ Thực hiện theo hướng dẫn sau
 - ❖ Nhập chuỗi nội dung, tìm kiếm và thay thế từ bàn phím
 - `String content = scanner.nextLine()`
 - `String find = scanner.nextLine()`
 - `String replace = scanner.nextLine()`
 - ❖ Thực hiện tìm và thay
 - `String result = content.replaceAll(find, replace)`

- ❑ Nhập chuỗi chứa dãy số phân cách bởi dấu phẩy và xuất các số chẵn
- ❑ Thực hiện
 - ❖ Sử dụng `split()` để tách chuỗi thành mảng bởi ký tự phân cách là dấu phẩy
 - ❖ Duyệt mảng, đổi sang số nguyên và kiểm tra số chẵn

```
String[] daySo = chuoi.split("")  
for(String so : daySo){  
    int x = Integer.parseInt(so);  
    if(x % 2 == 0){  
        Số chẵn  
    }  
}
```



LẬP TRÌNH JAVA 1

BÀI 7: CHUỖI VÀ BIỂU THỨC CHÍNH QUI PHẦN 2

❑ Bạn có biết các chuỗi sau đây biểu diễn những gì hay không?

❖ teo@fpt.edu.vn

❖ 54-P6-6661

❖ 54-P6-666.01

❖ 0913745789

❖ 192.168.11.200

1. Bạn có biết tại sao bạn nhận ra chúng không?
2. Làm thế nào để máy tính cũng có thể nhận ra như bạn?

- ❑ Máy tính có thể nhận dạng như chúng ta nếu chúng ta cung cấp qui luật nhận dạng cho chúng. Biểu thức chính qui cung cấp qui luật nhận dạng chuỗi cho máy tính.
- ❑ Biểu thức chính qui là một chuỗi mẫu được sử dụng để qui định dạng thức của các chuỗi. Nếu một chuỗi nào đó phù hợp với mẫu dạng thức thì chuỗi đó được gọi là so khớp (hay đối sánh).
- ❑ Ví dụ: **[0-9]{3,7}**: Biểu thức chính qui này so khớp các chuỗi từ 3 đến 7 ký tự số.
 - ❖ [0-9]: đại diện cho 1 ký tự số
 - ❖ {3,7}: đại diện cho số lần xuất hiện (ít nhất 3 nhiều nhất 7)

VÍ DỤ: BIỂU THỨC CHÍNH QUI

```
Scanner scanner = new Scanner(System.in);
```

```
System.out.print("Số mobile: ");  
String mobile = scanner.nextLine();
```

```
String pattern = "0[0-9]{9,10}";
```

```
if(mobile.matches(pattern)){  
    System.out.println("Bạn đã nhập đúng số mobile");  
}  
else{  
    System.out.println("Bạn đã nhập không đúng số mobile");  
}
```

Biểu thức
chính qui

Kiểm tra mobile có số
khớp với pattern không?

s.matches(regex)

Regular Expression

Ký tự đại diện

| | |
|-----------------|----------------------------------|
| [xyz] | đại diện một ký tự x, y hay z |
| [ad-f] | đại diện một ký tự a, d, e hay f |
| [^xyz] | đại diện ký tự không thuộc [xyz] |
| \d | tương đương [0-9] |
| \w | tương đương [0-9a-zA-Z_] |
| \D | tương đương [^\d] |
| \W | tương đương [^\w] |
| \s | đại diện ký tự trắng (\r\n\t\f) |
| . | đại diện ký tự bất kỳ |
| ^ | chỉ ra mẫu bắt đầu |
| \$ | chỉ ra mẫu kết thúc |
| \\, \., \\$, \^ | đại diện '\', '.', '\$' hay '^' |

Số lần xuất hiện

| | |
|-------|-----------------------------|
| {M,N} | Ít nhất M, nhiều nhất N lần |
| {N} | Đúng N lần |
| ? | 0-1 |
| * | 0-N |
| + | 1-N |
| Không | 1 |

[0-9]{3, 7}

❑ Số CMND

❖ `[0-9]{9}`

❑ Số điện thoại di động việt nam

❖ `0\d{9,10}`

❑ Số xe máy sài gòn

❖ `5\d-[A-Z]\d-((\d{4})|(\d{3}\.\d{2}))`

❑ Địa chỉ email

❖ `\w+@\w+(\.\w){1,2}`

```
Scanner in = new Scanner(System.in);

System.out.print("Email: ");
String email = in.nextLine();

System.out.print("Số điện thoại Huế: ");
String phone = in.nextLine();

String reEmail = "\\w+@\\w+\\.\\w+";
if(!email.matches(reEmail)) {
    System.out.println("Không đúng dạng email !");
}

String rePhone = "0543\\d{6}";
if(!phone.matches(rePhone)) {
    System.out.println("Không phải số điện thoại ở Huế !");
}
```

Email đơn giản

Số điện thoại để bàn ở Huế



DEMO

Hiện thực hóa đoạn mã ở slide trước



- ❑ Nhập thông tin nhân viên từ bàn phím. Thông tin của mỗi nhân viên phải tuân theo các ràng buộc sau. Xuất thông báo lỗi và yêu cầu nhập lại

| Thông tin | Kiểm soát | RegEx |
|--------------|----------------------------------|----------------------------------|
| Mã sinh viên | 5 ký tự hoa | [A-Z]{5} |
| Mật khẩu | Ít nhất 6 ký tự | .{6,} |
| Họ và tên | Chỉ dùng alphabet và ký tự trắng | [a-zA-Z]+ |
| Email | Đúng dạng email | \w+@\w+(\. \w+){1,2} |
| Điện thoại | Điện thoại Sài gòn | 083\d{7} |
| Số xe máy | Số xe máy Sài gòn | 5\d-[A-Z]-((\d{4}) (\d{3}\.{2})) |
| Số CMND | 10 chữ số | \d{10} |
| Website | Địa chỉ website | http://www\.\w+\.\w{2,4} |

- ☑ Giới thiệu chuỗi (String)
- ☑ Ký tự đặc biệt
- ☑ Thao tác chuỗi
- ☑ Giới thiệu biểu thức chính quy (Regular Expression)
- ☑ Xây dựng biểu thức chính quy
- ☑ Ứng dụng biểu thức chính quy

