

## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu *String*
2. Sử dụng class *StringBuilder*
3. Kỹ thuật xử lý chuỗi
4. *Regular Expression*
5. Kiểm tra sự hợp lệ của dữ liệu với *Regular Expression*
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. *BookStore Version 4*

## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu *String*
2. Sử dụng class *StringBuilder*
3. Kỹ thuật xử lý chuỗi
4. *Regular Expression*
5. Kiểm tra sự hợp lệ của dữ liệu với *Regular Expression*
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. *BookStore Version 4*

## Kiểu dữ liệu String

- *Tìm hiểu kiểu dữ liệu String*
- *Các trường hợp khởi tạo đối tượng kiểu String*
- *Chiều dài của chuỗi*
- *Nối chuỗi*
- *So sánh chuỗi == hoặc **equals** hoặc **equalsIgnoreCase***
- *Tạo ra vùng nhớ rác*



## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class **StringBuilder**
3. Kỹ thuật xử lý chuỗi
4. Regular Expression
5. Kiểm tra sự hợp lệ của dữ liệu với Regular Expression
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. BookStore Version 4

## Sử dụng class `StringBuilder`

- *Thao tác trên các kiểu dữ liệu `String` và không tạo ra vùng nhớ rác*
- *Các phương thức thông dụng*
  - *`append`*
  - *`insert`*
  - *`deleteCharAt`*
  - *`delete`*

## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class `StringBuilder`
3. **Kỹ thuật xử lý chuỗi**
4. Regular Expression
5. Kiểm tra sự hợp lệ của dữ liệu với Regular Expression
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. BookStore Version 4



## Kỹ thuật xử lý chuỗi cơ bản

- *length()* *Trả về độ dài chuỗi*
- *charAt()* *Lấy ra ký tự nằm ở vị trí index trong chuỗi*
- *toLowerCase()* *Chuyển chuỗi sang chữ thường*
- *toUpperCase()* *Chuyển chuỗi sang chữ in hoa*
- *trim()* *Loại bỏ khoảng trống 2 bên chuỗi*
- *substring()* *Trích xuất chuỗi con*

## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class *StringBuilder*
3. Kỹ thuật xử lý chuỗi
4. **Regular Expression**
5. Kiểm tra sự hợp lệ của dữ liệu với  
Regular Expression
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. BookStore Version 4



## Regular Expression là gì ?

- *Regular expresion (RE – biểu thức chính quy) bắt nguồn từ ngôn ngữ Perl, xuất hiện và được ứng dụng rất nhiều trong các ngôn ngữ lập trình*
- *Regular expression có thể hiểu đơn giản là một biểu thức dùng để mô tả một mẫu chuỗi nào đó (mẫu: các chuỗi văn bản tuân theo một quy luật sắp xếp nhất định).*
  - *dd/mm/yyyy là mẫu mô tả ngày tháng năm*
  - *xxxx@gmail.com là mẫu mô tả địa chỉ email*

## Regular Expression làm được gì ?

- *Biểu thức chính quy giúp giảm bớt những dòng mã trong quá trình xử lý chuỗi bằng những biểu thức ngắn gọn nhưng đem lại kết quả như mong đợi*
- *Regular expression thường được sử dụng trong các trường hợp*
  - *Kiểm tra giá trị các phần tử của Form*
  - *Xử lý các yêu cầu phức tạp trong chuỗi (bóc tách, thay đổi nội dung, loại bỏ ký tự, ...)*

## Ký hiệu thường dùng trong RE

STT	Ký hiệu	Ý nghĩa
1	^	Tìm kiếm giá trị từ đầu chuỗi nguồn
2	\$	Tìm kiếm giá trị từ cuối chuỗi nguồn
3	.	Đại diện cho một ký tự bất kỳ
4	\	Tìm kiếm các giá trị đặt biệt trong chuỗi (^ \$ . / @)
5	[]	Tìm tập hợp các ký tự
6	—	Lấy các ký tự trong một khoảng nào đó (thường dùng với ký hiệu [])
7	[^]	Nếu một ký tự ^ đứng trước một ký tự hay tập hợp có nghĩa là phủ định của ký hiệu hay tập hợp đó
8	(A B C)	Lựa chọn các giá trị A hoặc B hoặc C



## Ký hiệu thường dùng trong RE

STT	Ký hiệu	Ý nghĩa
9	*	0 hoặc nhiều lần xuất hiện
10	+	1 hoặc nhiều lần xuất hiện
11	?	0 hoặc 1 lần xuất hiện
12	{n,m}	Số lần xuất hiện của ký tự từ n đến m lần
13	*?	0 lần xuất hiện
14	+?	1 lần xuất hiện
15	??	0 lần xuất hiện
16	\w	Tương đương [A-z0-9_]
17	\W	Tương đương [^A-z0-9_]

## Ký hiệu thường dùng trong RE

STT	Ký hiệu	Ý nghĩa
18	\s	Tập hợp những ký tự khoảng trắng
19	\S	Tập hợp những ký tự không là ký tự khoảng trắng
20	\d	[0-9] Tập hợp những ký tự từ 0 đến 9
21	\D	[^0-9] Tập hợp những ký tự không thuộc từ 0 đến 9
22	\A	Tìm kiếm giá trị từ đầu chuỗi nguồn
23	\Z	Tìm kiếm giá trị từ cuối chuỗi nguồn

## Kiểm tra ID hợp lệ

1. ID phải có chiều dài là 7 ký tự *XXX-YYY*
2. *XXX* phải nằm trong tập hợp các giá trị ký tự A-Z hoặc a-z
3. *YYY* phải là các giá trị số từ 2 đến 8

**Pattern:** `[A-z]{3}-[2-8]{3}`



## Kiểm tra tên đăng nhập hợp lệ

1. Tên đăng nhập phải bắt đầu bằng một ký tự hoặc dấu gạch dưới
2. Tên đăng nhập là tập hợp của các ký tự a-z, 0-9 và có thể có các ký tự như dấu chấm ( . ), dấu gạch dưới ( \_ )
3. Độ dài tối thiểu của tên đăng nhập là 5 ký tự và độ dài tối đa là 32 ký tự

**Pattern** `[A-Za-z_][A-Za-z0-9_\.]{4,31}`

## Kiểm tra địa chỉ website hợp lệ

1. `http://www.zend.vn`
2. `https://www.zend.vn.vn`
3. `http://zend.vn`
4. `https://zend.vn`
5. `www.zend.vn`

**Pattern** `((https?:\//\//)(www\.)?|www\.)[a-z0-9\-\_]{3,20}(\.[a-z]{2,4}){1,2}`

## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu *String*
2. Sử dụng class *StringBuilder*
3. Kỹ thuật xử lý chuỗi
4. *Regular Expression*
5. Kiểm tra sự hợp lệ của dữ liệu với ***Regular Expression***
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. *BookStore Version 4*



## Kiểm tra sự hợp lệ của dữ liệu với RE

*Sử dụng phương thức matches để kiểm tra chuỗi có trùng khớp với mẫu RE hay không ?*

### ***Ví dụ***

- *Kiểm tra số tự nhiên hợp lệ*
- *Kiểm tra ID hợp lệ*
- *Kiểm tra username hợp lệ*
- *Kiểm tra địa chỉ website hợp lệ*

## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class StringBuilder
3. Kỹ thuật xử lý chuỗi
4. Regular Expression
5. Kiểm tra sự hợp lệ của dữ liệu với Regular Expression
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. BookStore Version 4

## Kỹ thuật tìm kiếm và thay thế chuỗi

- `replace` không kết hợp được với RE
- `replaceAll` kết hợp được với RE

### Bài tập áp dụng

- Java is v**o**ry v**o**ry **o**asy! → Java is v**er**y v**er**y **e**asy!
- kho**1**a ho**5**c la**65**p tri**2**nh java → khoa hoc lap trinh java
- **abcdAB69CD!@#\$%^&\*()**0<>?\_{}+21 → 69021



## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class `StringBuilder`
3. Kỹ thuật xử lý chuỗi
4. Regular Expression
5. Kiểm tra sự hợp lệ của dữ liệu với Regular Expression
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. **Kỹ thuật bóc tách chuỗi**
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. BookStore Version 4

## Kỹ thuật bóc tách chuỗi

- *token* chuỗi có giá trị xác định sau khi được bóc tách
- *delimiter* ký tự xác định dùng để bóc tách chuỗi

### Ví dụ

- *Java\_is\_easy!*
  - *Token: java, is, easy!*      *Delimiter: khoảng trắng*
- *Java--is--easy!*
  - *Token: java, is, easy!*      *Delimiter: --*

## Kỹ thuật bóc tách chuỗi

- *Scanner: useDelimiter, hasNext, next*
- *split*
- *StringTokenizer*

### Ví dụ

- *Java\_is\_very\_very\_easy*
- *Java--is--very--very--easy*
- *Java--is\_very\_very--easy*
- *Java\_12\_is\_233\_\_very\_32\_very\_\_156\_easy*



## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class `StringBuilder`
3. Kỹ thuật xử lý chuỗi
4. Regular Expression
5. Kiểm tra sự hợp lệ của dữ liệu với Regular Expression
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. BookStore Version 4

## Ứng dụng chuẩn hóa chuỗi

### ***Chuỗi đạt chuẩn dạng 1.***

- *Không có khoảng trắng ở đầu và cuối chuỗi*
- *Giữa các từ trong chuỗi chỉ tồn tại một khoảng trắng duy nhất*
- *Ký tự đầu tiên trong chuỗi phải là ký tự in hoa*
- *Các ký tự còn lại phải ở dạng chữ thường*
- *Hết câu bằng dấu chấm*

## Ứng dụng chuẩn hóa chuỗi

### ***Chuỗi Đạt Chuẩn Dạng 2.***

- *Không có khoảng trắng ở đầu và cuối chuỗi*
- *Giữa các từ trong chuỗi chỉ tồn tại một khoảng trắng duy nhất*
- *Ký tự đầu tiên trong chuỗi phải là ký tự in hoa,*
- *Hết câu bằng dấu chấm*
- *Các ký tự đầu tiên ở mỗi từ phải viết được viết hoa*



## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class `StringBuilder`
3. Kỹ thuật xử lý chuỗi
4. Regular Expression
5. Kiểm tra sự hợp lệ của dữ liệu với Regular Expression
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. **Ứng dụng chuyển số thành chữ**
10. Tình huống thực hành có đáp án
11. BookStore Version 4

## Ứng dụng đọc số có 3 chữ số

- $976 = \text{Chín trăm bảy mươi sáu}$
- $206 = \text{Hai trăm linh sáu}$
- $115 = \text{Một trăm mười lăm}$
- $291 = \text{Hai trăm chín mươi một}$

## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class *StringBuilder*
3. Kỹ thuật xử lý chuỗi
4. *Regular Expression*
5. Kiểm tra sự hợp lệ của dữ liệu với *Regular Expression*
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
- 10. Tình huống thực hành có đáp án**
11. *BookStore Version 4*



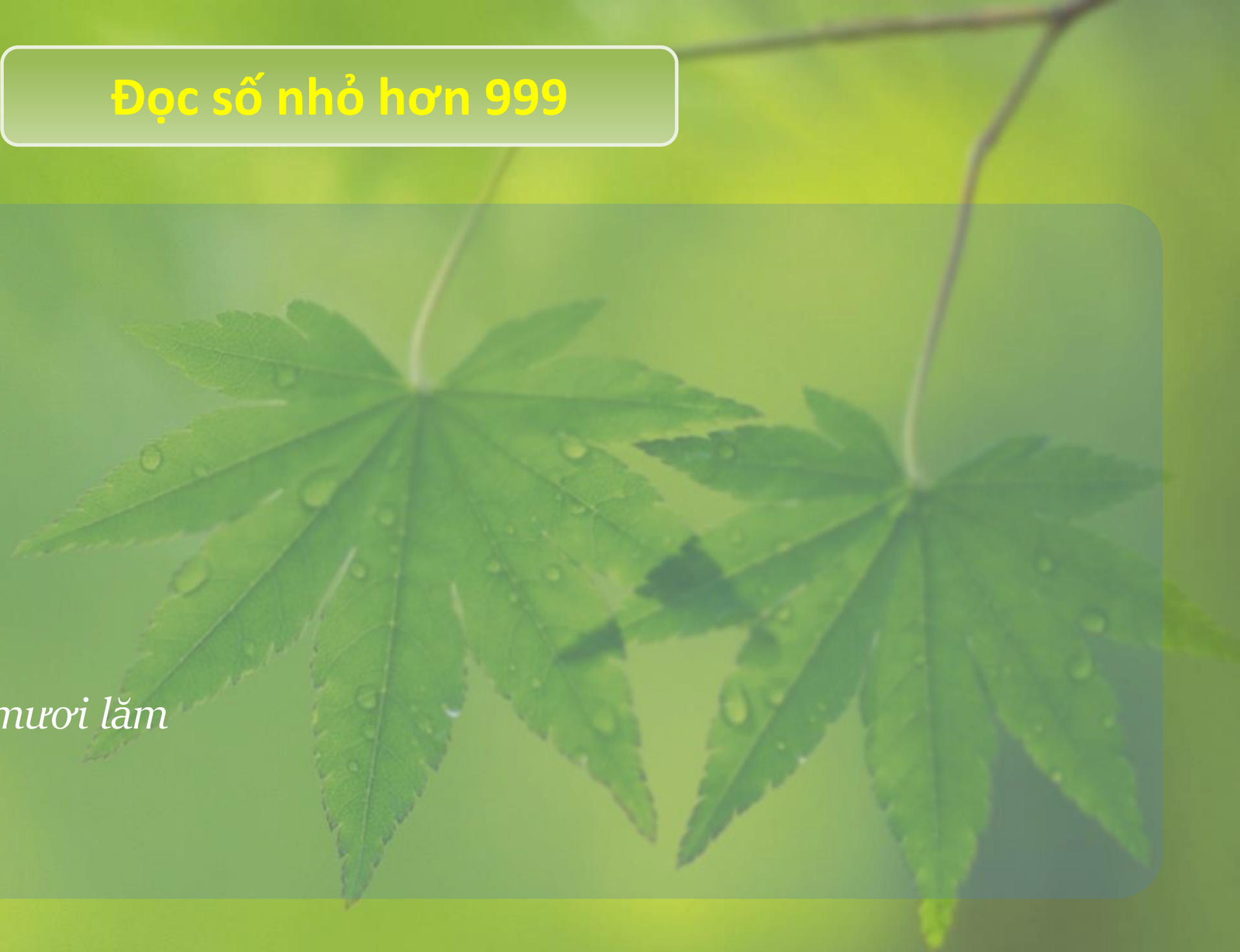
## Kiểm tra email hợp lệ

1. Địa chỉ email phải bắt đầu bằng một ký tự.
2. Địa chỉ email là tập hợp của các ký tự a-z, 0 đến 9 và có thể có các ký tự như dấu chấm ( . ), dấu gạch dưới ( \_ )
3. Độ dài tối thiểu của email là 5 ký tự và độ dài tối đa là 32 ký tự
4. Tên miền của email có thể là tên miền cấp 1 hoặc tên miền cấp 2

**Pattern:** `^[a-z][a-z0-9_\.]{4,31}@[a-z0-9]{2,}(\.[a-z0-9]{2,4}){1,2}$`

## Đọc số nhỏ hơn 999

- 9 chín
- 12 mười hai
- 97 chín mươi bảy
- 875 tám trăm bảy mươi lăm



## Đọc số có 12 chữ số

123.456.789.123

Một trăm hai mươi ba tỷ bốn trăm năm mươi sáu triệu bảy trăm tám mươi chín nghìn một trăm hai mươi ba đồng



## Chương 05:

# Kỹ thuật xử lý chuỗi

1. Kiểu dữ liệu String
2. Sử dụng class *StringBuilder*
3. Kỹ thuật xử lý chuỗi
4. *Regular Expression*
5. Kiểm tra sự hợp lệ của dữ liệu với *Regular Expression*
6. Kỹ thuật tìm kiếm và thay thế chuỗi
7. Kỹ thuật bóc tách chuỗi
8. Ứng dụng chuẩn hóa chuỗi
9. Ứng dụng chuyển số thành chữ
10. Tình huống thực hành có đáp án
11. **BookStore Version 4**