

NGÔN NGỮ LẬP TRÌNH JAVA

Chương 3 **CÁC LỚP TIỆN ÍCH TRONG JAVA (P1)**

NỘI DUNG

- 1. String**
- 2. StringBuffer**
- 3. StringBuilder**
- 4. String - Number casting**
- 5. Math**
- 6. Random**

NỘI DUNG

- 1. String**
- 2. StringBuffer**
- 3. StringBuilder**
- 4. String - Number casting**
- 5. Math**
- 6. Random**

Giới thiệu về String

- ❖ `java.lang.String`
- ❖ Mô tả các dữ liệu dạng chuỗi các ký tự kí tự cố định (immutable)
- ❖ Dữ liệu được khởi tạo cho chuỗi luôn là hằng số
- ❖ Ví dụ:
 - `String s1 = new String("Hello");`
 - `String s2 = "World";`

Các phương thức thông dụng của String

Phương thức	Mô tả
<code>int length()</code>	Trả về độ dài của chuỗi
<code>char charAt(int index)</code>	Trả về ký tự ở vị trí kí tự index trong chuỗi
<code>String trim()</code>	Loại bỏ các khoảng trắng hai đầu của chuỗi
<code>String replace(char oldChar, char newChar)</code>	Thay thế một ký tự trong chuỗi
<code>String substring(int start, int end)</code>	Trả về một chuỗi từ vị trí start đến vị trí end
<code>int compareTo(String s)</code>	So sánh chuỗi hiện tại với chuỗi s <ul style="list-style-type: none">- 0: nếu hai chuỗi bằng nhau- >0: nếu chuỗi s nhỏ hơn- <0: nếu chuỗi s lớn hơn
<code>int indexOf(String s)</code>	Tìm chuỗi s trong chuỗi hiện tại <ul style="list-style-type: none">- Trả về -1: nếu không tìm thấy- Trả về số nguyên n: nếu tìm thấy chuỗi s tại vị trí n
<code>static String valueOf(int value)</code>	Chuyển đổi giá trị kiểu dữ liệu đã cho thành chuỗi

Ví dụ

```
String s1 = "uit.", s2 = " edu.vn ", s3, s4;  
int i = s1.length();           // i = 4  
boolean b = s1.isEmpty();      // false  
char c = s1.charAt(i - 1);     // c = '.'  
s3 = s1.concat(s2);            // "uit. edu.vn "  
s4 = s1 + s2;                  // "uit. edu.vn "  
b = s3 == s4;                  // false - String is an object  
b = s3.equalsIgnoreCase(s4);  // true  
s3 = s4.substring(3);          // ". edu.vn "  
s3 = s4.substring(5, 8);       // "edu"  
s3 = s1 + s2;                  // "uit. edu.vn "  
b = s3.contains(s1);           // true  
b = s3.endsWith(s2);           // true  
b = s3.startsWith(s1);         // true  
b = s3.startsWith("edu", 5);  // true
```

Ví dụ

```
// s3 = "uit. edu.vn "  
// s2 = " edu.vn "  
  
i = s3.indexOf('u');           // 0  
i = s3.indexOf(s2);           // 4  
i = s3.indexOf("u", 0);       // 0  
i = s3.lastIndexOf("u", 2);   // 7  
s4 = s3.replace("t. ", "t."); // "uit.edu.vn "  
s4 = s3.trim();               // "uit. edu.vn"  
s4 = s3.toUpperCase();        // "UIT. EDU.VN"  
s2 = s4.toLowerCase();        // "uit. edu.vn"
```

NỘI DUNG

1. String
2. **StringBuffer**
3. StringBuilder
4. String - Number casting
5. Math
6. Random

Giới thiệu về StringBuffer

- ❖ `java.lang. StringBuffer`
- ❖ Được sử dụng để tạo chuỗi có thể thay đổi giá trị (mutable)
- ❖ Tương tự như lớp `String` ngoại trừ giá trị của nó có thể thay đổi
- ❖ Khi thực hiện nối nhiều chuỗi thì lớp `StringBuffer` xử lý nhanh và tốn ít bộ nhớ hơn (so với `String`)
- ❖ Ví dụ:
 - `StringBuffer s1 = new StringBuffer ("Hello");`
 - `StringBuffer s2 = "World";`

Các phương thức thông dụng của StringBuffer

Phương thức	Mô tả
<code>StringBuffer()</code>	Tạo ra một bộ đệm chuỗi với dung lượng 16 kí tự
<code>StringBuffer(String s)</code>	Tạo ra một bộ đệm chuỗi với chuỗi cho trước
<code>StringBuffer(int capacity)</code>	Tạo ra một bộ đệm chuỗi với dung lượng được chỉ định
<code>public synchronized StringBuffer append(String s)</code>	Nối thêm chuỗi s vào sau chuỗi hiện tại
<code>public synchronized StringBuffer insert(int offset, String s)</code>	Chèn chuỗi chỉ định với chuỗi này tại vị trí offset
<code>public synchronized StringBuffer replace(int startIndex, int endIndex, String s)</code>	Thay thế chuỗi từ vị trí startIndex đến endIndex bằng chuỗi s
<code>public synchronized StringBuffer delete(int startIndex, int endIndex)</code>	Xóa chuỗi từ vị trí startIndex đến endIndex
<code>public synchronized StringBuffer reverse()</code>	Đảo ngược chuỗi

Ví dụ

```
StringBuffer strbuff;  
strbuff = new StringBuffer("Ngon ngu lap trinh");  
System.out.println(strbuff.length()); //18  
strbuff.append("Java"); //Ngon ngu lap trinhJava  
strbuff.insert(18, " "); // Ngon ngu lap trinh Java  
strbuff.delete(0, 9); // lap trinh Java  
strbuff.reverse(); //avaJ hnirt pal
```

NỘI DUNG

1. String
2. StringBuffer
3. **StringBuilder**
4. String - Number casting
5. Math
6. Random

Giới thiệu về StringBuilder

- ❖ `java.lang.StringBuilder`
- ❖ Được sử dụng để tạo chuỗi có thể thay đổi (mutable)
- ❖ `StringBuilder` tương tự như lớp `StringBuffer` ngoại trừ nó không đồng bộ (non-synchronized) nên `StringBuilder` có thể cho hiệu năng cao hơn so với `StringBuffer`
- ❖ Ví dụ:
 - `StringBuilder s1 = new StringBuilder ("Hello");`
 - `StringBuilder s2 = "World";`

Các phương thức thông dụng của StringBuilder

Phương thức	Mô tả
<code>StringBuilder()</code>	Tạo ra một bộ đệm chuỗi với dung lượng 16 kí tự
<code>StringBuilder(String s)</code>	Tạo ra một bộ đệm chuỗi với chuỗi cho trước
<code>StringBuilder(int capacity)</code>	Tạo ra một bộ đệm chuỗi với dung lượng được chỉ định
<code>public StringBuilder append(String s)</code>	Nối thêm chuỗi s vào sau chuỗi hiện tại
<code>public StringBuilder insert(int offset, String s)</code>	Chèn chuỗi chỉ định với chuỗi này tại vị trí offset
<code>public StringBuilder replace(int startIndex, int endIndex, String s)</code>	Thay thế chuỗi từ vị trí startIndex đến endIndex bằng chuỗi s
<code>public StringBuilder delete(int startIndex, int endIndex)</code>	Xóa chuỗi từ vị trí startIndex đến endIndex
<code>public StringBuilder reverse()</code>	Đảo ngược chuỗi

Ví dụ

```
StringBuilder strbuild;  
strbuild = new StringBuilder("Java la ngon ngu");  
System.out.println(strbuild.length()); //18  
strbuild.append("lap trinh"); //Java la ngon ngulap trinh  
strbuild.insert(16, " "); // Java la ngon ngu lap trinh  
strbuild.reverse(); //hnirt pal ugn nogh al avaJ
```

Ví dụ

```
long startTime = System.currentTimeMillis();
StringBuffer strbuff = new StringBuffer("Hello, ");
for (int i = 0; i < 100000; i++)
    strbuff.append("Tôi là ngôn ngữ lập trình Java");
System.out.println("Thời gian nối chuỗi của StringBuffer: "
    + (System.currentTimeMillis() - startTime) + "ms");
startTime = System.currentTimeMillis();
StringBuilder strbuild = new StringBuilder("Hello, ");
for (int i = 0; i < 100000; i++)
    strbuild.append("Tôi là ngôn ngữ lập trình Java");
System.out.println("Thời gian nối chuỗi của StringBuilder: "
    + (System.currentTimeMillis() - startTime) + "ms");
```


NỘI DUNG

1. String
2. StringBuffer
3. StringBuilder
4. **String - Number casting**
5. Math
6. Random

Number -> String

❖ Phương thức toString()

- Integer.toString(123)
- Integer i = 123;
- i.toString();

❖ String.valueOf()

- String str = String.valueOf(123.456);

❖ Dùng phương thức format của String

- String str = String.format("%.3f", 123.456);

Number -> String

❖ Kết hợp phép cộng (+) chuỗi

- `float f = 1.2f;`
- `String str = "11" + f;` `//111.2`

❖ Với StringBuffer/StringBuilder, dùng phương thức append()

- `int i = 456;`
- `StringBuilder strbuild = new StringBuilder("123");`
- `strbuild.append(i);` `//"123456"`

String -> Number

- ❖ Dùng phương thức parse...() của lớp wrapper
 - `byte b = Byte.parseByte("12");`
 - `short s = Short.parseShort("32767");`
 - `double d = Double.parseDouble("2.5");`
- ❖ Dùng phương thức valueOf()
 - `String str="15";`
 - `int i=Integer.valueOf(str);`

NỘI DUNG

1. String
2. StringBuffer
3. StringBuilder
4. String - Number casting
5. Math
6. Random

Giới thiệu lớp Math

- ❖ `java.lang.Math`
- ❖ Là lớp tiện ích cung cấp các hàm về toán học
- ❖ Các hàm trong Math là static, để gọi hàm chỉ đơn giản viết tên lớp Math và tên phương thức cần gọi

Các hàm thông dụng

- ❖ **Math.PI:** hằng số PI `//3.141592653589793`
- ❖ **Math.abs(a):** trả về giá trị tuyệt đối của a
 - `int b = Math.abs(-2); // 2`
- ❖ **Math.ceil(a):** trả về giá trị double là số làm tròn tăng bằng giá trị số nguyên gần nhất với a
 - `double c = Math.ceil(1.234); // 2.0`
- ❖ **Math.floor(a):** trả về double là số làm tròn giảm
 - `double f = Math.floor(6.543); // 6.0`

Các hàm thông dụng

❖ **Math.max(a,b):** lấy số lớn trong hai số a, b

- `int SoLonHon = Math.max(10, 20); // 20`

❖ **Math.min(a,b):** lấy số nhỏ hơn

- `int m = Math.min(10, 20); // 10`

❖ **Math.pow(a,b):** lấy lũy thừa (cơ số a, số mũ b)

- `double p = Math.pow(3, 2); // 9.0`

❖ **Math.sqrt(a):** lấy căn bậc hai

- `double a = Math.sqrt(9); //3`

Các hàm thông dụng

❖ **Math.sin(a)/Math.cos(a):** sin và cos của góc đơn vị radian

- `double s = Math.sin(Math.PI/2);` //1

❖ **Math.random():** tạo số double ngẫu nhiên từ 0 đến 1

- `double r = Math.random();`

❖ **Math.toDegrees(double):** đổi góc radian thành độ

- `double goc = Math.toDegrees(Math.PI/2);` //90

❖ **Math.toRadians(double):** đổi góc đơn vị độ ra radian

- `double goc = Math.toRadians(30);` //0.5235987755982988

NỘI DUNG

1. String
2. StringBuffer
3. StringBuilder
4. String - Number casting
5. Math
6. Random

Random

- ❖ Sử dụng lớp Math: tạo số double ngẫu nhiên từ 0 đến 1
 - `double r = Math.random();`
- ❖ Sử dụng lớp Random
 - `Java.util.Random`
 - **`nextInt(int a)`**: tạo số ngẫu nhiên trong phạm vi từ 0 đến a (a số nguyên không âm)
 - `Random gen = new Random();`
 - `int value = gen.nextInt(4) + 1;` //tạo giá trị từ 1 - 4
 - `double value = gen.nextDouble() * 360;` //tạo giá trị góc 0 - 360 độ
 - `int value = gen.nextInt((max - min) + 1) + min;` // tạo giá trị nguyên trong khoảng min-max

Tóm tắt bài học

- ❖ String là lớp được sử dụng để tạo các chuỗi ký tự, các đối tượng được tạo ra có giá trị cố định (immutable)
- ❖ StringBuffer là lớp tương tự String, mutable, đồng bộ nhưng hỗ trợ thao tác có hiệu năng cao hơn
- ❖ StringBuilder là lớp tương tự StringBuffer nhưng không đồng bộ, có thể hỗ trợ thao tác có hiệu năng cao hơn StringBuffer
- ❖ Ép kiểu number -> String
 - toString()
 - String.valueOf()
 - String.format()

Tóm tắt bài học

- ❖ Ép kiểu String -> number
 - parse...()
 - Integer.valueOf(str)
- ❖ Math có nhiều phương thức hỗ trợ tính toán: **abs, ceil, floor, max, min, pow, sqrt, sin, random, toDegrees, toRadians,...**
- ❖ Random dữ liệu
 - Math.random(): tạo số ngẫu nhiên
 - Lớp Random: nextInt(int), nextDouble(),...