UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

NGÔN NGỮ LẬP TRÌNH JAVA

BUỔI 5
CÁC THÀNH PHẦN TIỆN ÍCH

GVGD: ThS. Lê Thanh Trọng

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

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Phương thức	Mô tả
int length()	Trả về độ dài của chuỗi
char charAt(int index)	Trả về ký tự ở vị trí kí tự index trong chuỗi
String trim()	Loại bỏ các khoảng trắng hai đầu của chuỗi
String replace(char oldChar, char newChar)	Thay thế một ký tự trong chuỗi
String substring(int start, int end)	Trả về một chuỗi từ vị trí start đến vị trí end
int compareTo(String s)	So sánh chuỗi hiện tại với chuỗi s - 0: nếu hai chuỗi bằng nhau - >0: nếu chuỗi s nhỏ hơn - <0: nếu chuỗi s lớn hơn
int indexOf(String s)	Tìm chuỗi s trong chuỗi hiện tại - Trả về -1: nếu không tìm thấy - Trả về số nguyên n: nếu tìm thấy choỗi s tại vị trí n
static String valueOf(int value)	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

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

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

Giới thiệu về StringBuffer

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

- 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

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

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

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

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

Giới thiệu về StringBuilder

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

- 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

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

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

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

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, dung 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

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

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 kểu number -> String
  - toString()
  - String.valueOf()
  - String.format()

Tóm tắt bài học

- Ép kểu 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(),…

Bài tập

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

1. Viết chương trình nhập vào 2 chuỗi. Xuất ra chiều dài của hai chuỗi và xây dựng phương thức so sánh 2 chuỗi bằng cách duyệt qua lần lượt từng ký tự của chuỗi, thực hiện việc so sánh 2 chuỗi theo hướng dẫn:
   a. Chuỗi s1 lớn hơn chuỗi s2 khi s1 có kích thước lớn hơn s2: trả về 1
   b. Nếu s1 và s2 có kích thước bằng nhau lần lượt xét từng cặp ký tự tương ứng từ trái sang phải. Nếu s1[i] > s2[i] trả về kết quả là 1, Nếu s1[i] < s2[i] trả về kết quả là -1.
   c. Nếu s1 và s2 có kích thước bằng nhau và tương ứng từng cặp kí tự đều bằng nhau thì trả về kết quả là 0
2. Viết chương trình cho phép nhập vào một đoạn văn bản và xuất ra:
   a. Số ký tự của văn bản (không tính các dấu câu: chấm (.), hỏi (?), chấm cảm (!), phẩy (,), chấm phẩy (;).
   b. Xuất ra số từ của văn bản.
   c. Xử lý các khoảng trắng thừa (đầu và cuối đoạn, hai khẳng trắng liên tục).
   d. Chuẩn hóa đoạn văn bản nếu kí tự đầu đoạn hoặc sau dấu chấm câu không được viết hoa.
3. Viết chương trình cho phép nhập vào một từ và xuất ra nguyên âm, phụ âm của từ đó. Ví dụ “thương” có nguyên âm là “th”, phụ âm là “ương”.

Bài tập

4. Viết chương trình nhập vào danh sách các họ tên của N người. Xuất danh sách theo thứ tự tăng dần của tên (alphabet).
5. Viết chương trình sử dụng số ngẫu nhiên để tạo câu. Cài đặt 4 mảng các String là "article", "noun", "ver"", "prepositon". Tạo các câu bằng cách lựa chọn ngẫu nhiên mỗi từ trong mảng, có định dạng: article-noun-verb-preposition-article. Các từ trong câu cách nhau bằng một dấu gạch. In ra 10 câu ngẫu nhiên từ các mảng như sau:
   - article: the, an a, one, some, any.
   - noun: boy, girl, man, dog, car, town.
   - verb: drove, jumped, ridden, walked, kicked, hit.
   - preposition: to, from, over, on, under."
6. Số nguyên lớn là là số có số lượng ký số lên đến hàng chục, hàng ngàn. Xây dựng lớp số nguyên lớn và các phương thức +, -, *, / giữa hai số lớn. Viết chương trình cho phép tạo ra hai số nguyên lớn (n và m ký số) vào hai số nguyên lớn và xuất ra kết quả tổng, hiệu, tích, thương giữa hai số nguyên lớn vừa nhập.