CS202 Java Object Oriented Programming

Strings

Chengyu Sun
California State University, Los Angeles

String

- A sequence of characters enclosed in a pair of double quotes
  - “abcd”, “\ “ abcd\””, “\\\\”, “” …
- Create a String

```
String sa = “abcd”;
String sb = new String( a );

char[] c = { ‘a’, ‘b’, ‘c’, ‘d’ };
String sc = new String ( c );
```

http://java.sun.com/j2se/1.4.2/docs/api/java/lang/String.html

String Methods – General

- length()
- charAt()

```
String s = “abcd”;

// . note the difference between string length and
//   array length
// . character index range
for( int i=0 ; i < s.length() ; ++i )
     System.out.println( s.charAt(i) );
```

String Methods – Comparison

- equals(String)
- equalsIgnoreCase(String)
- compareTo(String)
- compareToIgnoreCase(String)

```
String s1 = “abcd”;
String s2 = new String( “abcd” );
String s3 = “bcde”;

System.out.println( s1 == s2 );           // ??
System.out.println( s1.equals(s2) );      // ??
System.out.println( s1.compareTo(s3) ); // ??
```

String Methods – Search

- Search for either a character or a string
- `indexOf` – search from left to right
- `lastIndexOf` – search from right to left

```
String s4 = “ababa”;

System.out.println( s4.indexOf(‘a’) );          // ??
System.out.println( s4.lastIndexOf(‘a’) );      // ??
System.out.println( s4.indexOf(‘a’,3) );        // ??
System.out.println( s4.lastIndexOf(‘a’,3) );    // ??
System.out.println( s4.indexOf(“ab”) );         // ??
System.out.println( s4.lastIndexOf(“ab”) );     // ??
```

String Methods – Substring

- Extract substring from a string
  - substring(int beginIndex)
  - substring(int beginIndex, int endIndex)

```
String s4 = “ababa”;

String subs1 = s4.substring( 1 );        // ??
String subs2 = s4.substring( 1, 3 );     // ??
```

String Methods – Manipulation

- toUpperCase()
- toLowerCase()
- replace(char oldChar, char newChar)
- replaceAll(String oldStr, String newStr)
- trim()

```
String s5 = “    Object Oriented Programming    ”;

System.out.println( s5 );
System.out.println( s5.toUpperCase() );
System.out.println( s5.toLowerCase() );
System.out.println( s5.trim() );
```

Strings are Immutable

- Once a String object is created, it cannot be changed
- Use StringBuffer class if you *really* need a mutable string

```
String a = “abcd”;
String b = a;
String c = new String( a );
String d = “abcd”;

System.out.println( a == b );
System.out.println( a == c );
System.out.println( a == d );

a += “cde”;
System.out.println( a );
System.out.println( b );
System.out.println( a == b );
```

StringTokenizer

- Breaks a string into substrings, called “tokens”.
  - “123 10 101” => “123” “10” “101”
- Delimiters
  - Default: white spaces
  - Or could be anything
- Important methods
  - Constructors
  - `hasMoreTokens`, `nextToken`, and `countTokens`

StringTokenizer Example

```
import java.util.StringTokenizer;
…

String s = “123 10 101”;
int a[];

StringTokenizer st = new StringTokenizer( s );
a = new int[ st.countTokens() ];

int index=0;
while( st.hasMoreTokens() )
  a[index++] = Integer.parseInt( st.nextToken() );
```

StringTokenizer Exercise

- Tokenize a string delimited by both commas and white spaces, such as “128, 10, 101”

A Faster Way to Split a String

- String[] split(String regex)
  - Regex is the *regular expression* of the delimiter

```
String s = “123 10 101”;
String tokens[] = s.split(“ ”);
```