

CS202 Java Object Oriented Programming Strings, Characters, and Wrapper Classes

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"

Character Class

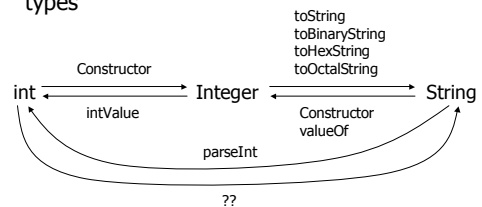
- ◆ Constructor
 - Character(char c)
- ◆ Static test methods
 - isLetter(char), isDigit(char), isLetterOrDigit(char)
 - isUpperCase(char), isLowerCase(char)
 - isWhitespace(char)

Wrapper Classes of Primitive Types

- ◆ For each primitive type there's a corresponding class
 - boolean – Boolean
 - int – Integer
 - char – Character
 - double – Double
 - ...
- ◆ Provide some utility functions for a certain primitive type

Integer

- ◆ Constants
 - MAX_VALUE and MIN_VALUE
- ◆ Methods for conversions among different types



Integer Example

```

Integer a = new Integer(10);           // int to Integer
int d = a.intValue();                 // Integer to int
int c = Integer.parseInt("1234");     // String to int

String bin = Integer.toBinaryString(a); // Integer to String
String hex = Integer.toHexString(a);   // Integer to String
String oct = Integer.toOctalString(a); // Integer to String

Integer b = new Integer("10");        // String to Integer
Integer n = Integer.valueOf("101");   // String to Integer
Integer m = Integer.valueOf("101", 2); // String to Integer
    
```

Lab Assignment Revisited

- ◆ Given an integer number consists of only 0's and 1's, get its binary, octal, and hexadecimal values

```

ConsoleReader in = new ConsoleReader();
int a = in.readInt();

String s = "" + a; // int to String

int v2 = Integer.valueOf(s, 2).intValue();
int v8 = Integer.valueOf(s, 8).intValue();
int v16 = Integer.valueOf(s, 16).intValue();
    
```

Confusion in the Textbook

- ◆ D&D v5, Chapter 10.10, p485
 - "Each of the type wrappers is declared final, so their methods are implicitly final and may not be overridden"
- ◆ final classes cannot be inherited, period