

Java Enterprise Application Development

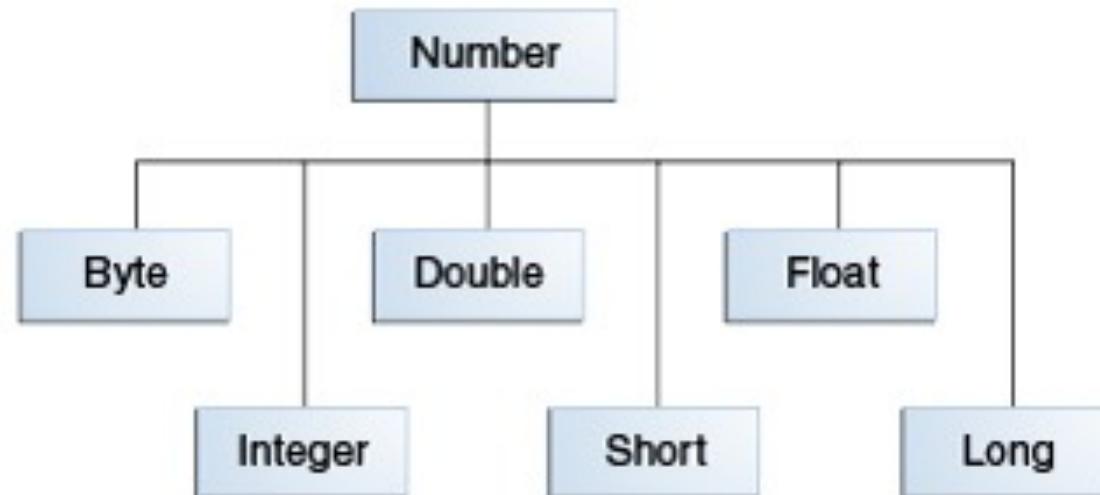
Lecture 3 Numbers and Strings

**Dr. Fan Hongfei
22 September 2025**

Wrapper Classes

- There are reasons to use **objects in place of primitives**
 - Java platform provides **wrapper classes**

Primitive type	Wrapper class
boolean	Boolean
byte	Byte
char	Character
float	Float
int	Integer
long	Long
short	Short
double	Double



Autoboxing and Unboxing

- Wrapping can be done by the compiler

```
Character ch = 'a';
```

- **Autoboxing**

- Passed as a parameter to a method that expects an object of the corresponding wrapper class
 - Assigned to a variable of the corresponding wrapper class

- **Unboxing**

- Passed as a parameter to a method that expects a value of the corresponding primitive type
 - Assigned to a variable of the corresponding primitive type

Methods Implemented

- Converting the value of Number object to the primitive data type returned
- Comparing the Number object to the argument
- Determining whether the Number object is equal to the argument
- Conversion (with *Integer* as example)
 - `parseInt(String s)`, `parseInt(String s, int radix)`
 - `toString()`
 - `valueOf(int i)`

Formatting Numeric Print Output

- `print`
- `println`
- `public PrintStream format(String format, Object... args)`
 - Format specifiers begin with a percent sign (%) and end with a converter
- The `DecimalFormat` class
 - For your private study

Beyond Basic Arithmetic

- The Math class
 - Math.E, Math.PI
 - abs
 - ceil, floor, round
 - min, max
 - exp, log, pow, sqrt
 - sin, cos, tan, asin, acos, atan
- Random numbers
 - The random() method returns a pseudo-randomly selected number between 0.0 and 1.0

Characters

- The Character class
 - `isLetter`, `isDigit`
 - `isWhitespace`
 - `isUpperCase`, `isLowerCase`
 - `toUpperCase`, `toLowerCase`
 - `toString`
- Escape sequences
 - `\t`, `\b`, `\n`, `\r`, `\'`, `\'',` `\\"`

Strings

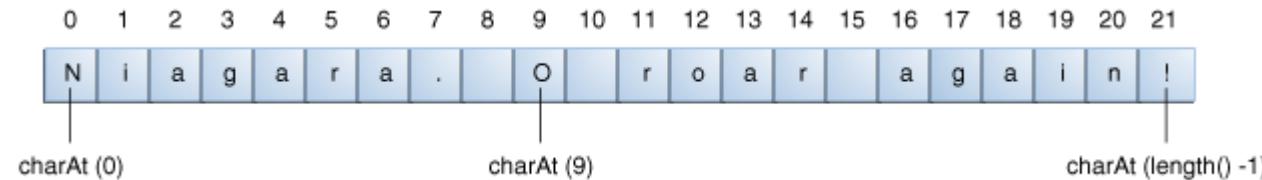
- The String class has 13 constructors that allow you to provide the initial value
- The String class is **immutable**
 - Once created, a String object cannot be changed
- `length()`
- `concat()`
- `format()`

Converting between Numbers & Strings

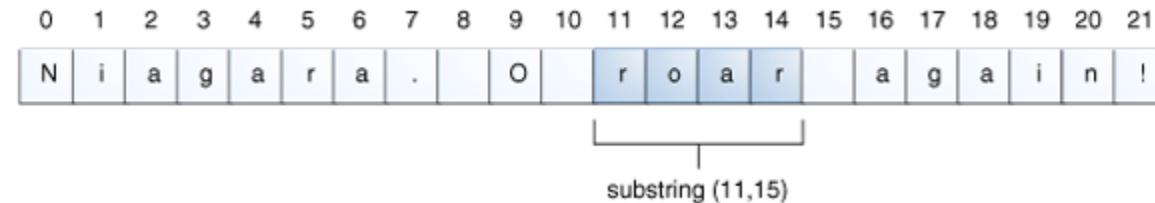
- Converting Strings to Numbers
 - A method `valueOf` converts a string to an object
 - Each of the `Number` subclasses that wrap primitive numeric types also provides a `parseXXX()` method
- Converting Numbers to Strings
 - `String.valueOf()`
 - `toString()`

Manipulating Characters in a String

- `charAt`



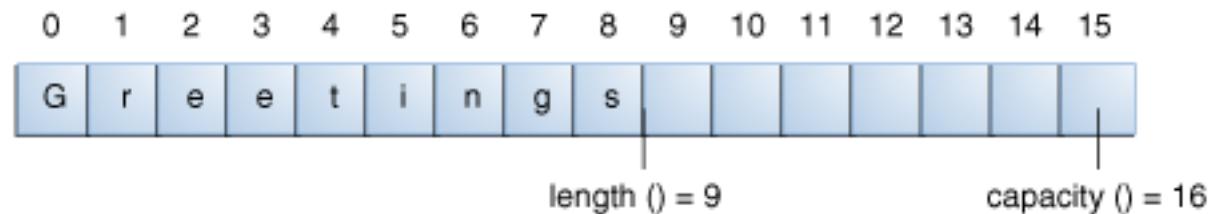
- `substring`



- `trim`, `toLowerCase`, `toUpperCase`
- `indexOf`, `lastIndexOf`, `contains`
- `replace`, `replaceAll`, `replaceFirst`

The StringBuilder Class

- StringBuilder objects are like String objects, except that they can be modified
- `StringBuilder()`, `StringBuilder(int initCapacity)`



- `append`, `delete`, `deleteCharAt`, `insert`
- `replace`, `setCharAt`, `reverse`
- `toString`

Practice

1. Implement a program, which accepts a list of strings provided by the user, and outputs the sorted list of strings ordered by length. For strings with the same length, they are sorted by alphabetical order.
2. Implement a program, which allows the user to input a string (without space), and outputs the list of characters (such as letters, digits, symbols) occurred in the string. The same letters in lowercase and uppercase are regarded as a unique one. The characters are sorted by their occurrences in the string.