

String

a sequence of characters, implemented as array

create using 1) constructor new String("") 2) literals 3)concatenation

empty string ""

Concatenation

if anyone of operand is string, all other operands are cast to String

toString() for reference type

primary is convert directly to string

If none of operand is String, the return will not be string

int x=3;y=4; String s=x+y; // error can not assign int to String

Default toString() return memory address for Object class: class name@memoryaddress

Compare String

== memory location, only for test

equals();equalsIgnoreCase() content

compareTo() content

Immutability and String Pool

Once initialized, never changed again, Chain of methods results in multiple Strings

String literals could be a largest chunk of memory.

String pool

string literals,literal concatenation (compiler time)

obj.intern(): will return a string in string pool or added if not exist

String s = "He" + lo;//runtime concatenation, not using string pool

Math, Wrapper class, String: immutable
Math,Wrapper, String, StringBuilder, System: final

Methods

length() method instead of attribute, trade-offspace and performance

charAt() StringIndexOutOfBoundsException

indexOf() 0..length()-1, -1

substring(begin,end) [...], exception, empty string

toLowerCase() toUpperCase()

startsWith() endsWith()

s1.contains(s2) s.indexOf(s2) !=-1

replace(oldChar,newChar)

replace(oldCharSeq,newCharSeq) character sequence - String, StringBuilder

trim() trim white space

StringBuidler Methods (cont)

substring(int), substring(int,int) returns string type

charAt();indexOf();length();substring() see String class

equals() is not overridden in StringBuilder, use toString().equals(s1);

Why StringBuilder

efficient memory usage

Mutable and chain of operation will result in one sb

before java5, StringBuffer was used, but slower due to thread safe.

Create a StringBuilder

new StringBuilder();

new StringBuilder(sb2);

new StringBuilder(size)=0; capacity=5
r(5);

StringBuidler Methods

append(str)

insert(offset,str)

delete(from,to)

deleteCharAt()

reverse()

toString() sb for performance, convert to string in the end