5.Strings

Sakib Abrar

CSE

Bangladesh University of Engineering & Technology

sakib.cghs@gmail.com

September 8, 2020

Overview

String related classes

String basics

String constructors

String length and character extraction

Extracting Substrings

String Comparisons

String Concatenation

String Search

String Split

String Conversions

String to other type conversions

String related classes

- Java provides three String related classes
- java.lang package
  –String class: Storing and processing Strings but Strings created using the String class cannot be modified immutable
  –StringBuffer class: Create flexible Strings that can be modified
- java.util package
  –StringTokenizer class: Can be used to extract tokens from a String

String Basics

- String class provide many constructors and more than 40 methods for examining in individual characters in a sequence.
- You can create a String from a String value or from an array of characters.
  –String newString = new String(stringValue);
- The argument stringValue is a sequence of characters enclosed inside double quotes
  –String message = new String (“Welcome”);
  –String message = “Welcome”;

String constructors

StringConstructors.java

```
1  public class StringConstructors {
2      public static void main(String[] args) {
3          char charArray[ ] = { 'b',  'i',  'r',  't',  'h',  ' ',  'd',  'a',  'y'} ;
4          byte byteArray[ ] = { ( byte)  'n',  ( byte)  'e',  ( byte)  'w', ( byte)  ' ',
5                  ( byte)  'y',  ( byte)  'e',  ( byte)  'a',  ( byte)  'r'} ;
6
7          String s = new String( original: "hello"); // hello
8          String s1 = new String() ; //
9          String s2 = new String(s) ; // hello
10         String s3 = new String(charArray); // birth day
11         String s4 = new String(charArray,  offset: 6,  count: 3); // day
12         String s5 = new String(byteArray,  offset: 4,  length: 4); // year
13         String s6 = new String(byteArray); // new year
14         String s7 = "Wel"  +  "come"; // Welcome
15
16         System.out.println(s);
17         System.out.println(s1);
18         System.out.println(s2);
19         System.out.println(s3);
20         System.out.println(s4);
21         System.out.println(s5);
22         System.out.println(s6);
23         System.out.println(s7);
24     }
25 }
26
```

String length & character extraction

- Returns the length of a String
  –length()
- Example:
  String s1=“Hello”;
  System.out.println(s1.length);

Extraction

- Get the character at a specific location in a string
  –s1.charAt(1)
- Get the entire set of characters in a string
  –s1.getChars(0, 5, charArray, 0)

Extracting Substrings

- substring method enable a new String object to be created by copying part of an existing String object
  –substring (int startIndex) - copies the characters form the starting index to the end of the String
  –substring(int beginIndex, int endIndex) - copies the characters from the starting index to one beyond the endIndex

String Comparisons

- equals
  –Compare any two string objects for equality using lexicographical comparison. s1.equals(“hello”)
- equalsIgnoreCase
  –s1.equalsIgnoreCase(s2)
- compareTo
  –s1.compareTo(s2)
  –s1 ¿ s2 (positive), s1 ¡ s2 (negative), s1 = s2 (zero)

String Concatenation

- Java provide the concat method to concatenate two strings.
  String s1 = new String ("Happy ");
  String s2 = new String ("Birthday");
  String s3 = s1.concat(s2);
  s3 will be "Happy Birthday"

String Search

- Find the position of character/String within a String
  –int indexOf (char ch)
  –int lastIndexOf (char ch)

String Split

- split() method splits a String against given regular expression and returns a character array
- String test = "abc,def,123";
  String[] out = test.split(",");
  out[0] - abc , out[1] - def, out[2] - 123

String Conversions

- Generally, the contents of a String cannot be changed once the string is created,
- Java provides conversion methods
- **toUpperCase()** and **toLowerCase()**
  –Converts all the characters in the string to lowercase or uppercase
- **trim()**
  –Eliminates blank characters from both ends of the string
- **replace(oldChar, newChar)**
  –Replaces a character in the string with a new character

String to other type conversions

The String class provides **valueOf** methods for converting a character, an array of characters and numeric values to strings
–**valueOf** method take different argument types

Type	To String	From String
boolean	String.valueOf(boolean)	Boolean.parseBoolean(String)
byte	String.valueOf(int)	Byte.parseByte(String, int base)
short	String.valueOf(int)	Short.parseShort (String, int base)
Int	String.valueOf(int)	Integer.parseInt (String, int base)
long	String.valueOf(long)	Long.parseLong (String, int base)
float	String.valueOf(float)	Float.parseFloat(String)
double	String.valueOf(double)	Double.parseDouble(String)

THE END