Java API’s Comparisons Bonus

Michael Colistro

COMP 4413

DR SABAH MOHAMMED

Scanner API:

Text scanner that parses primitive types and strings using regular expressions. What it does is it breaks the input into tokens using a delimiter pattern which matches whitespace. The tokens can be converted into values of different types using a next method. The following example is how the user inputs a Long with System:

Scanner myscan = new Scanner(System.in);

long myLong = myscan.nextLong();

String Tokenizer:

This breaks a string into tokens. Its much simpler than StreamTokenizer. It is unable to distinguish between different variable types and comments. The delimiters can be specified at creation time or on a per-token basis. StringTokenizer behaves one of two ways which depends if the returnDelims flag is set or not. If its false the delimiter characters separate the tokens. IF it is true the delimiter character are considered to be tokens aswell. It also keeps a current position within the string to be tokenized. A token gets returned by taking a substring. This is an example on how to use STringTokenizer:

StringTokenizer st = new StringTokenizer("This is for Sabahs Bonus!");

while (st.hasMoreTokens()) {

System.out.println(st.nextToken());

}

The output would be:

This

Is

For

Sabahs

Bonus!

StreamTokenizer:

This takes an input stream and parses it into tokens. The parsing is controlled by a table and it uses flags that determine various states. The stream can identify different variable types andcomments. Each byte read is regarded as a character in the range of ‘\u0000’ through ‘\u00FF’. It can look up five different attributes of the character: white space, alphabetic, numeric, string quote, and comment character. Every char has to be one of these attributes. The flags previously spoken about indicate the following:

If line terminators should be returned a tokens or treated as white space

If c-style comments are to be recognized or skipped

If c++ style comments are to be recognized or skipped

If the characters of identifiers are voncerted to lower case

This is an example of StreamTokenizer:

mystring = reader.readLine();

token=new StreamTokenizer((new StringReader(mystring)));

int numOfTokens = token.nextToken();

while(numOfTokens!= token.TT\_EOF)

{

System.out.println(token.sval);

numOfTokens = token.nextToken();

}

String.split():

This is a built in string function. When called must be passed and argument of type char to specify where the string must be split. To use special characters aswell split(‘’\\.’’) must be called to split on the period. String.split() is also extremely slow and performance is just generally bad. It returns an array of strings which will be more clear in the following example which shows how to use.

String string = "This is-an example";

String[] str = string.split("-");

System.out.println(str[0]);

System.out.println(str[1]);

The output would be:

This is

an example

Pattern.split():

This splits text into an array of strings aswell like the string.split which uses the regular expression as the delimiter. What it does is takes the String pattern and looks for that instance in the regular string and splits up everytime it finds that pattern. The tokens then go into the array like string.split

String text = "This cat is cat for cat Sabahs cat Bonus";

String patternString = "cat";

Pattern pattern = Pattern.compile(patternString);

String[] split = pattern.split(text);