Lecture 8: Strings and string operations

Jan 30 2015

(We’ll start with the Scanner method Tuesday)
char data type review

• A char type variable consists of 2 bytes
• It contains the Unicode code of some character
• A char type variable is an integer type variable that contains a positive number
  – Consider which are “safe” and “unsafe” conversations
Strings

- A string is a sequence of characters enclosed between the double quotes "..."

Example:
- "abc123"
- "Hello World"
- "Hello, what is your name ?"

Each character in a string is of the type char and uses the Unicode encoding method
**String** is NOT a *primitive* data type

- The **String** data type is the first data type that we learn that is not built-in into the programming language Java
- Primitive data types of built-in data types of Java:
  - `double`
  - `float`
  - `byte`
  - `short`
  - `int`
  - `long`
  - `char`
  - `boolean` (we will learn this soon)
The **String** data type is a *class*

- The **String** data type is a class that is constructed using the char data type:
- A *data type* has:
  - information
  - Operations
  Example: the integer data type contains a value (= information) and operations (such as: +, -, etc.)
- Implementing the **String** data type in Java:
  - The information of the **String** data type is stored using a number of *char* typed variables
  - The operations on **String** data are implemented by a number of methods (i.e. computer programs)
  - The documentation of the class **String** (you can find a list of the methods) can be found at [http://docs.oracle.com/javase/6/docs/api/java/lang/String.html](http://docs.oracle.com/javase/6/docs/api/java/lang/String.html)
For now…

• we don’t have enough knowledge of Java to explain how the **String** data type is constructed. (It uses advanced program language knowledge.)

• We will learn about this later in the course

• For now, we will only learn how to *use* the **String** data type
String literals

- We write string literals between double quotes “...” Example: "Hello World"

- An *escape character* is a special character that allows Java to change the meaning of the *next* character:
  \ (backslash) = the escape character for strings

- An *escape sequence* is the escape character \ followed by *one* character. Example: \n
  - An escape sequence denotes one character
  - The character denoted by an escape sequence is usually one that you cannot type in with the keyboard
## Commonly used escape sequences

<table>
<thead>
<tr>
<th>Escape sequence</th>
<th>Denoted character</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>\t</td>
<td>Tab</td>
<td>&quot;\t&quot;</td>
</tr>
<tr>
<td>\n</td>
<td>New line</td>
<td>&quot;\n&quot;</td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>Backslash ()</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>&quot;&quot;</td>
<td>Double quote</td>
<td>&quot;&quot;&quot;</td>
</tr>
</tbody>
</table>
How strings are stored

• Challenge to store strings:
  Strings have variable lengths !!!

Example:
  • "Hello" contains 5 characters
  • "Hello\n" contains 6 characters
  • "Good–bye" contains 8 characters

Conclusion: You cannot store a string in a fixed sized variable !!!

• There are 2 techniques in Computer Science for storing strings:
  – The location + length method
  – The location + sentinel method
Location + length method

• The characters of string is stored (using Unicode encoding) somewhere *consecutively* inside the RAM memory

• A string variable contains:
  – the location of the first character of the string
  – the length of the string (total number of characters in the string)
Storing strings "abc" and "12" using location + length
Location + sentinel method

• Sentinel = a special character (symbol) that denotes the end of a string

• The sentinel used in programming languages to denote the end of a string is usually the NULL character with Unicode value 0.

• The location + sentinel method:
  – The characters of string is stored (using Unicode encoding) somewhere consecutively inside the RAM memory
  – The sentinel (NULL character) is added after the last character of the string
  – A string variable contains:
    • the location of the first character of the string
    • (The end of the string is marked by the sentinel)
Storing strings "abc" and "12" using location + sentinel
Which method is better

- According to this article, the location + length method is much more efficient

http://www.mathcs.emory.edu/~cheung/Courses/170/Syllabus/05/Docs/Null-string.pdf
Strings in Java

• Java uses the location + length method to store String typed data
• The expression `a.length()` returns the length of the string stored in the string variable `a`
Numbers vs. numeric strings

Integer numbers and numeric strings look visually alike in a computer program:

<table>
<thead>
<tr>
<th>Java code</th>
<th>What's stored inside computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 (integer number)</td>
<td>000000000 00000000 00000000 00001100</td>
</tr>
<tr>
<td>&quot;12&quot; (numeric string)</td>
<td>00000000 00110001 00000000 00110010</td>
</tr>
<tr>
<td></td>
<td>^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^</td>
</tr>
<tr>
<td></td>
<td>Unicode for '1'  Unicode for '2'</td>
</tr>
</tbody>
</table>

Notice that:

• What is stored inside the computer are completely different
• (because they use different representation (encoding) techniques)
The + operator on a number typed variable and a string typed variable

When the + operator is used between a number and a string

Then

– the number is automatically converted to a string
– The + operator is then applied on 2 strings (i.e., the + operator is a concatenation !)