Lecture Notes: Esc 101

Dt. 8th Feb., 2008

Strings

String is a class in Java. Java library has various classes. We can include specific classes in our program by using the import and class name.
E.g. import java.lang.* includes all the classes under java.lang.

However, by default this is already included and thus when we specify String, we need not include the class.

String s = new String();
s is an object of class String created by calling constructor String() with empty string.

String s = new String("Esc101");
This defines a new string s with initialization value Esc101.

String t = new String(s); … (1)
String t = s; ……. (2)
There is a difference between (1) and (2).
In (1) we create a memory box and the characters are copied from s.
If s is modified then t doesn't change.
In contrast, in (2) both refer to the same memory box. So, if s is changed, t changes too.

Operations:

1. Comparing
   (s = = t)
   To compare if s & t are equal. However, the corresponding memory locations of s and t are fetched and then compared.

   E.g. s-> ABC and t-> ABC, (s= =t) will evaluate to false.

   s->t-> "ABC", will evaluate to true.

2. '+' operator
   This operator is used for appending one string to another.
   s = s +'X';
   will result in s->"ABCX" if initial value of s was "ABC".

String s = "Esc101"; …….(3)
String s = new String("Esc101"); ….(4)
In case of (3) the compiler creates a memory box having constant value "Esc101".
In case of (4) the compiler converts the argument to string and is then passed to the String constructor to create a new memory box.

String s,t;
s=”ABC”;
t=s;
Here s and t refer to the same memory box.

E.g. String s = new String(“ABC”);
t=s;
s=”XYZ”;
Here, first a new memory box having “ABC” is created which is assigned to s.
Later a reference to this memory box is stored in t.
On the third line, the compiler creates a memory box having constant value “XYZ” and s now refers to this box.
Thus, in the end, s has value “XYZ” and t has “ABC”.