



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica  etsinf

La classe String

Introducció a la Informàtica i la Programació (IIP)
Curs 2019/20

Departament de Sistemes Informàtics i Computació



La classe String

Capítol 5 – Secció 5.1 del llibre de l'assignatura.

- La classe `String` es una classe predefinida (dintre del paquet `java.lang`) mitjançant la qual es poden manipular cadenes de caràcters.

Referència: <http://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

- Els seus objectes es poden construir de diferents formes, però qualsevol grup de caràcters entre cometes dobles és una referència a un objecte d'eixe tipus.

Per exemple:

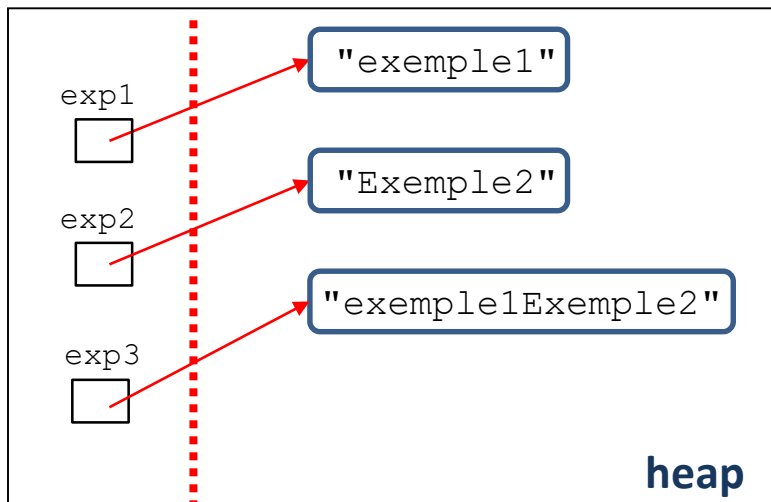
```
String st1 = "Això és un exemple de String";  
String st2 = new String("Això és un exemple de String");  
String st3 = "";    // String buit, que NO null!!  
String st4 = new String(); // Ídem
```

La classe String

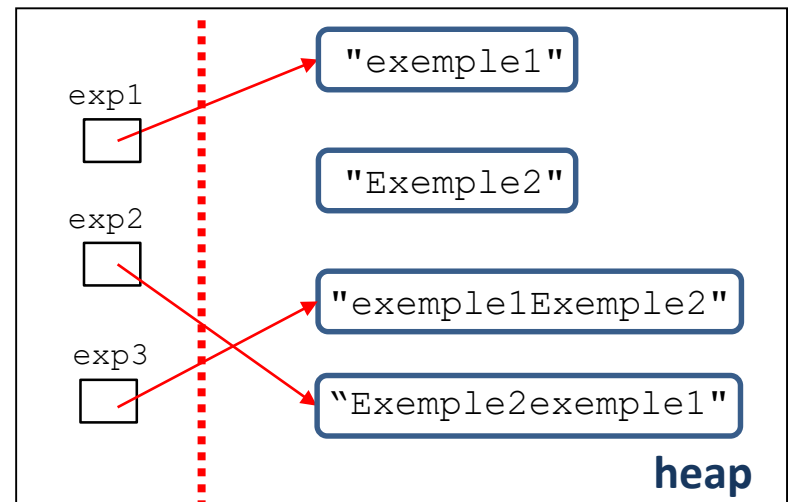
- Els String són **immutables**, és a dir, una vegada inicialitzats no és possible alterar el seu contingut (afegint, eliminant o canviant caràcters).
- La **concatenació** (afegir un String després d'un altre, **creant un nou**) és l'operació d'ús més habitual per generar nous String a partir d'altres. L'operador de concatenació és el "+", podent-se emprar l'operador "+=".

```
String exp1 = "exemple1";  
String exp2 = "Exemple2";  
String exp3 = exp1 + exp2; // exp3 referència a "exemple1Exemple2" (1)  
exp2 += exp1;              // exp2 referència a "Exemple2exemple1" (2)
```

Memòria del sistema (1)



Memòria del sistema (2)



La classe String. Comparacions

- Els operadors "==" i "!=" **comparen referències**, no els objectes *String* que representen.
- Els operadors relacionals ">", ">=", "<", "<=" no estan definits i no es poden usar per a comparar valors *String*.
- Per a la **comparació d'igualtat** es fa servir el mètode `equals(Object)`, així:

```
boolean b = st1.equals(st2);  
// b és true si st1 i st2 tenen exactament els mateixos caràcters.
```

- Per a la **comparació d'ordre** s'usa el mètode `compareTo(String)`.

```
int i = st1.compareTo(st2); // on l'enter i és:  
// menor que 0 si st1 és anterior a st2.  
// major que 0 si st1 és posterior a st2.  
// igual a 0 si st1 és igual caràcter a caràcter a st2.
```

La comparació de cadenes es basa en l'**ordre lexicogràfic**. És a dir, les cadenes es comparen entre si com en un diccionari, però tenint en compte el seu codi.

La classe String. Comparacions

Exemples:

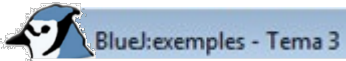
```
String s1 = "Hola", s2 = "Hello", s3;  
boolean iguals;  
  
iguals = s1 == s2;           // iguals == false  
iguals = s1.equals(s2);      // iguals == false  
  
s3 = s1;  
iguals = s3 == s1;           // iguals == true  
iguals = s3.equals(s1);      // iguals == true  
  
int comp1 = s3.compareTo(s1); // comp1 == 0  
int comp2 = s3.compareTo(s2); // comp2 > 0  
int comp3 = s2.compareTo(s3); // comp3 < 0
```

La classe String. Mètodes rellevants

Mètode	Significat
length()	Torna la llargària (nombre de caràcters) de la cadena.
trim()	Torna la cadena eliminant els espais en blanc al principi i al final.
charAt(n)	Torna el caràcter que està en la posició <i>n</i> .
substring(ini, fi) substring(ini)	Torna la subcadena compresa entre les posicions <i>ini</i> i <i>fi-1</i> . Sense <i>fi</i> , torna fins a la fi de la cadena.
toUpperCase()	Torna la cadena amb les minúscules convertides a majúscules.
toLowerCase()	Torna la cadena amb les majúscules convertides a minúscules.
indexOf(cad)	Torna la posició de la primera aparició de <i>cad</i> en la cadena o -1 si no existeix.
lastIndexOf(cad)	Torna la posició de l'última aparició de <i>cad</i> en la cadena o -1 si no existeix.
startsWith(prefixe)	Torna true sii la cadena comença per <i>prefixe</i> .
endsWith(sufixe)	Torna true sii la cadena acaba amb <i>sufixe</i> .
concat(cad)	Torna la cadena concatenant al final <i>cad</i> .
contains(seq)	Torna true sii la cadena conté la seqüència <i>seq</i> .

NOTA: Els mètodes que tornen cadenes, retornen **cadenaes noves**, diferents de les originals.

La classe String. Exemples de mètodes



- **Executa** en el **CodePad de BlueJ** del projecte les instruccions següents, **mostrant** el seu **valor** i **comprovant** que coincideix amb l'indicat en els comentaris.

```
String st1 = "Exemple 1";
               0 1 2 3 4 5 6 7 8

String majus = st1.toUpperCase();           // majus és "EXEMPLE 1"
String minus = st1.toLowerCase();           // minus és "exemple 1"
int llargària = st1.length();                // llargària == 9
char caràcter = st1.charAt(1);               // caràcter == 'x'
String sub = st1.substring(3, 5);            // sub és "mp"
String st = st1.concat(" i 2");              // st és "Exemple 1 i 2"
boolean b = st1.startsWith("Exe");          // b == true
boolean c = st1.endsWith("Exe");            // c == false
int inici = st1.indexOf("mpl");              // inici == 3
int desde = st1.indexOf("mpl", 2);           // desde == 3

String st2 = "  Exemple 2  ";
               0 1 2 3 4 5 6 7 8 9 10 11 12 13
String noBlanc = st2.trim();                 // noBlanc és "Exemple 2"
int ultima = st2.lastIndexOf("  ");         // ultima == 12
               0 1
```