Lenguaje de programación I

CIBERTEC

- Observa las siguientes líneas de código:

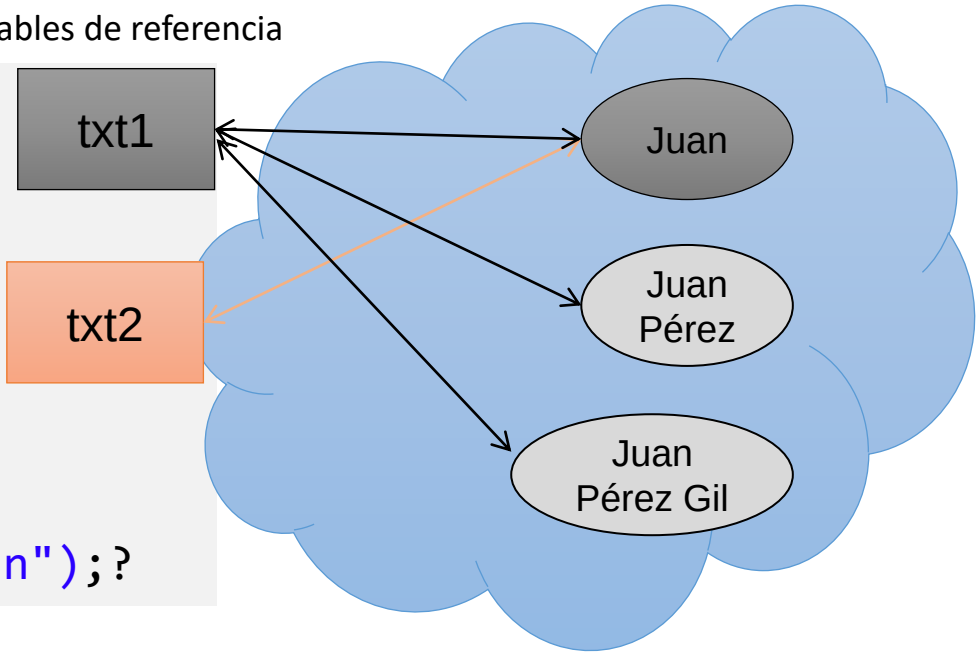
```
String txt1 = "Juan";

String txt2 = "Juan";

txt1 += "Pérez";

txt1 += "Gil";

¿String txt3 = new String ("Juan");?
```



- Al "cambiar" el valor de una cadena, en realidad se compara su existencia apuntando a la **dirección** del objeto existente o creando un nuevo objeto.

¿Qué consecuencias puede tener esto para el desarrollo de aplicaciones?

Contenido

- La clase String
  - Definición y gestión de memoria
  - Principales métodos de la clase String
- Las clases StringBuffer y StringBuilder

Logros de la Unidad

- El alumno crea aplicaciones utilizando de manera individual o combinada las clases String, StringBuffer, StringBuilder

La clase String

- Cuando el compilador encuentra un literal String, verifica el pool para validar si ya existe una cadena idéntica, si la encuentra, la referencia al nuevo literal es dirigida a dicha cadena evitando la creación de un nuevo objeto String.

- La clase String es inmutable, por lo que una vez que se crea un objeto String no se puede cambiar. Quiere decir que cuando operamos con cadenas estas se comparan en el " “, apunta a las direcciones de los objetos contenidos.
- Esto último es una de las razones por las cuales String es una **clase final**.

StringBuffer y StringBuilder

- Cuando, tenemos que realizar **varias modificaciones**, a los textos podemos usar las clases **StringBuffer** y **StringBuilder** .
- Los objetos de tipo StringBuffer y StringBuilder pueden ser modificados una y otra vez sin generar objetos abandonados de tipo String.

StringBuffer.
•Es menos rápido. •Es sincronizado (Multihilos). •Es seguro.

StringBuilder.
•Es más rápido. •No son sincronizados. •Es menos seguro.

Comparando Clases

```
long REPETICIONES = 60000; // veces a evaluar


String s = "";
long tiempo = System.currentTimeMillis(); // tiempo inicial
for (int i = 0; i < REPETICIONES; i++) {
   s += "a";
}
tiempo = System.currentTimeMillis() - tiempo;
System.out.println("String ha tardado: " + tiempo + " milisegundos");


StringBuffer s2 = new StringBuffer();
tiempo = System.currentTimeMillis();
for (int i = 0; i < REPETICIONES; i++) {
   s2.append("a");
}
tiempo = System.currentTimeMillis() - tiempo;
System.out.println("StringBuffer ha tardado: " + tiempo + " milisegundos");
```

Métodos de la clase String

- Se pueden emplear métodos encadenados. Ej. method1().method2();

isEmpty()	Devuelve true o false si el String está vacío o no.
length()	Devuelve el número de caracteres de un String
equals (cad2) **equalsIgnoreCase(**cad2)	Compara la cadena con el parámetro
charAt(pos).	Devuelve el caracter en el índice especificado**.**
toLowerCase() **toUpperCase()**	Convierte un String a minúscula o mayúscula respectivamente
startsWith(cad) endsWith(cad)	Devuelve true o false el texto inicia o empieza con la cadena indicada

Métodos de la clase String

•String **trim().**	Devuelve una cadena eliminando los espacios en blanco al inicio o final
replace(coriginal, cnuevo**).**	Reemplaza las ocurrencias de un caracter por otro**.**
split(separador).	Devuelve un array de String, separando elementos según el parámetro separador.
indexOf(cad, inicio)	devuelve la posición de primera aparición de una cadena dentro de un String, por defecto busca desde el principio
substring(*inicio, final*)	Retorna parte de un String
concat(cadena)	Agrega un String al final de otro. (similar a +)

Propuesto

- Dado el siguiente texto:

  "a001; José ; Atúncar ;12;15;LP1"

- Usa los métodos de la clase String para:
  - Separar los textos en código, nombre, apellido, nota1, nota2 y curso
  - Convertir el apellido a mayúscula y el nombre el minúscula
  - Validar que el curso sea LP1
  - Validar que el código empieza con la letras “a” o “p” y que el tamaño no sea mayor a 4
  - Generar un correo con el siguiente formato:
    1° letra del nombre + parte numérica del código + @cibertec.pe. Ej. j001@cibertec.pe
  - Mostrar los datos

StringBuffer y StringBuilder

length()	Devuelve el número de caracteres (tamaño) del texto
append(cad)	Añade la cadena al String
delete(inicio, fin)	Elimina el texto indicado en el intervalo
insert(inicio, cad)	Inserta en la posición indicada la cadena
reverse()	Devuelve el texto en forma inversa
charAt(pos).	Devuelve el caracter en el índice especificado**.**
setCharAt(pos, caracter).	Establece un caracter en el índice especificado
toString()	Convierte a String el contenido
indexOf(cad, inicio)	devuelve la posición de primera aparición de una cadena dentro de un String, por defecto busca desde el principio

Aplicación 3

- Dado una cadena, realizar un método que “encripte” las letras del texto considerando lo siguiente:
  a->e, e->i, i->o, o->u, u->a

Problems | Javadoc | Declaration | Console

<terminated> DemoCadenas2 [Java Application] C:\Program Files\Jav

```
>>> CiberEncriptador <<<
Texto original : Hola mundo esto es un texto
Texto encriptado : utxit na si utsi udnam eluH
>>> CiberDesEncriptador <<<
Texto original : utxit na si utsi udnam eluH
Texto desencriptado : Hola mundo esto es un texto
```

Referencia

- http://www.slideshare.net/jorg_marq/clase10-stringsio

GRACIAS

SEDE MIRAFLORES
Calle Diez Canseco Cdra 2 / Pasaje Tello
Miraflores – Lima
Teléfono: 633-5555

SEDE INDEPENDENCIA
Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA
Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO
Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO
Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE SAN MIGUEL
Av. Federico Gallese 847
San Miguel – Lima
Teléfono: 632-4900

SEDE BELLAVISTA
Av. Mariscal Oscar R. Benvides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA
Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535