



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)**

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
Ingeniero en Computación	2009-1	12099	Programación Orientada a Objetos

PRÁCTICA No.	LABORATORIO DE	Programación Orientada a Objetos	DURACIÓN (HORAS)
7	NOMBRE DE LA PRÁCTICA	Conversión entre tipos de datos y la clase String	2

1. INTRODUCCIÓN

En esta práctica comprobaremos el funcionamiento de las clases String, Integer, Float y Double. La clase String es utilizada para manipular cadenas y contiene muchos métodos para realizar diversas acciones con las cadenas. Las clases Integer, Float y Double se utilizan para manipular objetos que contienen valores de tipos int, float y double respectivamente. Estas clases también contienen métodos para convertir cadenas a un valor del tipo primitivo correspondiente.

2. OBJETIVO (COMPETENCIA)

Aplicar la conversión entre tipos de datos y la manipulación de cadenas mediante las clases String, Integer, Float y Double.

Formuló Cecilia Curlango Rosas María Luisa González Ramírez	Revisó Gloria E. Chávez Valenzuela	Aprobó	Autorizó M.C. Maximiliano de las Fuentes Lara
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Responsable de Gestión de Calidad	Nombre y Firma del Director de la Facultad

3. FUNDAMENTO

La clase `String` se utiliza para la manipulación de cadenas, dentro de ella se crea un arreglo de caracteres parecido a lo que pasa en el lenguaje C. El arreglo se puede manipular utilizando los métodos que contiene la clase `String`.

Podemos crear objetos `String` de forma explícita o implícita. De forma implícita se crea cuando ponemos una serie de caracteres entre comillas dobles “ ”. Un ejemplo de esto es cuando mandamos a mostrar en la consola un mensaje de la siguiente forma:

```
System.out.println("Hola, esto es una cadena");
```

De forma explícita hay varias formas de crear el objeto `String`, por ejemplo:

```
String str= new String("Hola, esto es una cadena");  
String otraCadena =" Hola esto es otra cadena";
```

Java define el operador `+` (suma) con un significado especial cuando los operandos son de tipo `String`. En este caso el operador suma significa concatenación. El resultado de la concatenación es un nuevo `String` compuesto por las dos cadenas, una tras otra. Por ejemplo:

```
String nombre = "Maria" + "Gonzalez";
```

El valor de `nombre` será “MariaGonzalez”.

Es posible concatenar a un `String` datos primitivos. Por ejemplo, se puede usar:

```
int i = 5;  
String cadena = "El valor de i es " + i;
```

A continuación se muestran algunos de los métodos de la clase `String`

Tipo retorno	Método	Descripción
int	<code>length()</code>	Devuelve la longitud de la cadena que es el número de caracteres que contiene.
char	<code>charAt(int index)</code>	Devuelve el carácter en la posición indicada por <code>index</code> . El rango de <code>index</code> va de 0 a <code>length() - 1</code> .
String	<code>toLowerCase()</code>	Devuelve la cadena transformada a minúsculas.
String	<code>toUpperCase()</code>	Devuelve la cadena transformada a mayúsculas.
int	<code>compareTo(String otro)</code>	Compara dos cadenas, la del objeto con la que recibe como argumento. Regresa un valor entero mayor que 0 si es mayor la del objeto, igual a 0 si son iguales o menor que 0 si es menor la del objeto.

int	compareToIgnoreCase(String otro)	Compara dos cadenas, la del objeto con la que recibe como argumento, ignorando las diferencias entre letras mayúsculas y minúsculas. Regresa un valor entero mayor que 0 si es mayor si es mayor la del objeto, igual a 0 si son iguales o menor que 0 si es menor si es mayor la del objeto.
boolean	equalsIgnoreCase(String otro)	Verifica si dos cadenas son iguales.
int	indexOf(int ch)	Regresa el índice de la primera aparición del ch en la cadena.
String []	split(String v1)	Divide la cadena en un arreglo de Strings de acuerdo al valor de v1.
boolean	startsWith(String pre)	Verifica si la cadena comienza con el valor especificado por pre.

Además de los métodos anteriores, la clase `String` tiene otros métodos para manipular cadenas que son de gran utilidad. Para conocer cuales métodos tiene `String` y su funcionamiento se debe consultar el API de la clase `String` en:

<http://download.oracle.com/javase/6/docs/api/java/lang/String.html>

Las clases `Integer`, `Float`, `Double` y `Long` se le conoce como clases envoltorio o *wrappers* debido a que contienen o pueden manipular valores de tipos primitivos de datos. Estas clases contienen métodos para realizar la conversión de cadena al tipo primitivo correspondiente.

La clase `Integer` se utiliza para manipular valores de tipo entero. Envuelve un valor entero en un objeto. Un objeto de la clase `Integer` solo contiene un atributo que es de tipo `int`. Además provee de métodos para convertir un `int` a `String` y de `String` a `int` así como otros métodos que son útiles cuando manejamos un valor `int`.

La clase `Integer` cuenta con dos constructores para crear un objeto. Podemos crear un objeto con el valor `int`.

```
Integer ob=new Integer(5);
```

También podemos crear un objeto utilizando una cadena.

```
Integer ob2=new Integer("123");
```

Algunos de los métodos de la clase se describen a continuación.

Tipo retorno	Método	Descripción
static int	parseInt(String s)	Convierte la cadena a su valor entero
int	intValue()	Regresa el valor del int almacenado en el objeto Integer.
int	compareTo(Integer obj)	Compara un objeto interno Integer contra obj. Regresa un valor entero mayor que 0 si es mayor el objeto interno que el que recibe como argumento, igual a 0 si son iguales o menor que 0 si es menor el objeto interno que el que recibe como argumento.
double	doubleValue()	Regresa el valor del objeto como un double.
float	floatValue()	Regresa el valor del objeto como un flotante.
static String	toBinaryString(int i)	Regresa la representación en binario del entero i.
static String	toHexString(int i)	Regresa la representación en hexadecimal del entero i.
static String	toOctalString(int i)	Regresa la representación en octal del entero i.

La clase Long tiene métodos con un funcionamiento similar a los de la clase Integer por lo cual no se describirán en esta práctica. Sin embargo es recomendable consultar el API de ambas clases para conocer con mayor claridad el funcionamiento de ambas clases. Se puede consultar el API para Integer y Long en: <http://download.oracle.com/javase/6/docs/api/java/lang/Integer.html>
<http://download.oracle.com/javase/6/docs/api/java/lang/Long.html>

La clase Float se utiliza para manipular valores de tipo flotante. Envuelve un valor flotante en un objeto. Un objeto de la clase Float solo contiene un atributo que es de tipo float. Además provee de métodos para convertir un float a String y de String a float así como otros métodos que son útiles cuando manejamos un valor float.

La clase Float cuenta con tres constructores para crear un objeto. Podemos crear un objeto con el valor float.

```
Float ob=new Float(5);
```

y de forma similar con un valor double.

También podemos crear un objeto utilizando una cadena.

```
Float ob2=new Float("1.23");
```

Algunos de los métodos de la clase se describen a continuación.

Tipo retorno	Método	Descripción
static float	parseFloat(String s)	Convierte la cadena a su valor flotante
float	floatValue()	Regresa el valor flotante de este objeto.
int	intValue()	Regresa el valor de este objeto como un int
long	longValue()	Regresa el valor del objeto como un long

La clase `Double` tiene métodos con un funcionamiento similar a los de la clase `Float` por lo cual no se describirán en esta práctica. Sin embargo es recomendable consultar el API de ambas clases para conocer con mayor claridad el funcionamiento de ambas clases. Se puede consultar el API para `Float` y `Double` en:

<http://download.oracle.com/javase/6/docs/api/index.html?java/lang/Long.html>

<http://download.oracle.com/javase/6/docs/api/index.html?java/lang/Float.html>

4. PROCEDIMIENTO (DESCRIPCIÓN)

1.- Realice un programa que de la línea de comando lea dos cadenas. Utilice los métodos necesarios para cambiarlos a valores numéricos enteros y realice la suma de los dos números. Muestre en pantalla el resultado.

2.- Realice un programa que lea de la línea de comando una cadena y verifique si es un correo electrónico.

3.- Realice un programa que lea de la línea de comando una cadena con la forma `cadena1:cadena2:cadena3:cadena4`. Esta cadena la debe separar utilizando los dos puntos “:” como separador. Mostrar las cadenas en orden inverso.

```
cadena4
cadena3
cadena2
cadena1
```

4.- Considere la siguiente cadena:

```
String hannah = "Did Hannah see bees? Hannah did.";
```

Conteste las siguientes preguntas:

- ¿Cuál es el valor desplegado por la expresión `hannah.length()`?
- ¿Cuál es el valor devuelto por la llamada de método `hannah.charAt(12)`?
- Escriba una expresión que se refiera a la letra b en la cadena referida por `hannah`.

5.- Escriba un programa que identifique sus iniciales a partir de su nombre completo y las despliegue.

6. -Realice un programa que lea dos argumentos de la línea de comando, el primero deberá ser un número, el segundo cualquiera de las siguiente opciones B, H, O. Su programa deberá convertir el número a su representación indicada por el segundo valor, de acuerdo a la siguiente tabla.

Opción	Conversión
B	convierte el numero a su representación binaria
H	convierte el numero a su representación en hexadecimal
O	Convierte el número a su representación a octal

Por ejemplo:

```
java Convierte 23 B
```

el resultado sería:

```
10111
```

Otro ejemplo:

```
java Convierte 45 H
```

el resultado sería:

```
2D
```

A) EQUIPO NECESARIO	MATERIAL
----------------------------	-----------------

Computadoras con capacidad para ejecutar el entorno de desarrollo Netbeans. Paquete `misClases`. Al menos una imagen para manipularla con los métodos presentados y desarrollados en la práctica.

7. REFERENCIAS

Netbeans

<http://netbeans.org/downloads/>

Java 6

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>