

Boris Molina Igor  
Taller de programación II  
[bmolintai@ing.ucsc.cl](mailto:bmolintai@ing.ucsc.cl)

# Reglas del laboratorio

- Se propone tener un ejercicio evaluado cada semana.
- La modalidad de los ejercicios será con jueces virtuales.

# Imports

```
import java.util.ArrayList;  
import java.util.List;  
import java.util.Collections;  
import java.util.Scanner;
```

# String

- Los **String** en Java son vectores de caracteres, pero su manipulación no es como un vector tradicional como vieron en C.
- Los **String** en Java NO son **mutables**.
- **StringBuffer** y **StringBuilder** son un objetos de tipo **String**, pero su característica es que estos si son **mutables**.

API Java String: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

# Métodos

```
String s = "Hola Mundo!";
```

Tipo	Método	Descripción
boolean	<code>equals(String aString)</code>	Compara este String con el objeto especificado.  <code>s.equals("Hola Mundo!")</code> //Retorna true si son iguales los String, false caso contrario
int	<code>length()</code>	Devuelve la longitud de este String.  <code>s.length()</code> //Retorna la longitud del String
boolean	<code>isEmpty()</code>	Verifica si está vacío.  <code>s.isEmpty()</code> //Retorna true si su largo es 0, false caso contrario

# Métodos

```
String s = "Hola Mundo!";
```

Tipo	Método	Descripción
boolean	<code>startsWith(String suffix)</code>	Verifica si este String empieza con el <b>sufijo</b> especificado.
<code>s.startsWith("Hola") //Retorna true si empieza con ese sufijo, false caso contrario</code>		
boolean	<code>endsWith(String suffix)</code>	Verifica si este String termina con el <b>sufijo</b> especificado.
<code>s.endsWith("Mundo!") //Retorna true si termina con ese sufijo, false caso contrario</code>		
String	<code>replace(String oldS, String newS)</code>	Devuelve un String resultante de reemplazar todas las apariciones de <b>oldS</b> en este String por <b>newS</b> .
<code>s.replace("Hola", "Hello") //Reemplaza el primer String por el segundo String</code>		

# Métodos

```
String s = "Hola Mundo!";
```

Tipo	Método	Descripción
int	<code>indexOf(String str)</code>	Devuelve el índice dentro de este String de la primera aparición del subString especificado.
<code>s.indexOf("Hola") //Retorna el índice de la primera aparición</code>		
int	<code>lastIndexOf(String str)</code>	Devuelve el índice dentro de este String de la última aparición del subString especificado.
<code>s.lastIndexOf("Mundo!") //Retorna el índice de la última aparición</code>		
int	<code>hashCode()</code>	Devuelve un código Hash para este String.
<code>s.hashCode() //Retorna un valor int para un Hash</code>		

# Métodos

```
String s = "Hola Mundo";
```

Tipo	Método	Descripción
char	<code>charAt(int index)</code>	Devuelve el char del índice especificado.
<code>s.charAt(0) //Retorna el carácter en ese índice</code>		
String	<code>concat(String str)</code>	Concatena el String especificado al final de este String.
<code>s.concat("ooooo!") //Retorna un String concatenado</code>		
boolean	<code>contains(String str)</code>	Verifica si contiene este String.
<code>s.contains("Hol") //Retorna true si contiene el String, false caso contrario</code>		



# Métodos

```
String s = "Hola Mundo";
```

Tipo	Método	Descripción
String	<code>substring(int begin, int end)</code>	Devuelve un String que es un subString de este String.
<code>s.substring(0, 3) //Retorna un subString según los índices</code>		
String	<code>toLowerCase()</code>	Convierte todos los caracteres de este String a minúsculas.
<code>s.toLowerCase() //Retorna un String convertido en minúsculas</code>		
String	<code>toUpperCase()</code>	Convierte todos los caracteres de este String a mayúsculas.
<code>s.toUpperCase() //Retorna un String convertido en mayúsculas</code>		

# Métodos

```
String s = "Hola Mundo";
```

Tipo	Método	Descripción
String[]	<code>split(String regex)</code>	Divide este String entre coincidencias de la expresión dada.
<pre>String []vec = s.split(" "); //Retorna un vector con todos los subString separados ["Hola", "Mundo"]</pre>		
char[]	<code>toCharArray()</code>	Convierte este String en un vector de caracteres.
<pre>char []vec = s.toCharArray(); //Retorna un vector con todos los caracteres separados [H, o, l, a, , M, u, n, d, o]</pre>		
String	<code>valueOf(Object anObject)</code>	Devuelve un String del objeto.
<pre>String str = String.valueOf(anObject); //Convierte a String un Objeto dado</pre>		

# StringBuilder

```
StringBuilder sb = new StringBuilder();
```

Tipo	Método	Descripción
StringBuilder	<code>append(String str)</code>	Agrega el String especificado a este StringBuilder.
<code>sb.append("Hola Mundo"); //Agrega un String al final del StringBuilder</code>		
StringBuilder	<code>insert(int index, String str)</code>	Inserta el String especificado en el índice del StringBuilder.
<code>sb.insert(4, "Hola Mundo"); //Inserta un String en el índice del StringBuilder</code>		

# Competencia 1

# Indicaciones Competencia 1 (Evaluada)

1. Entrar a [vjudge.net](http://vjudge.net).
2. Registrarse en la página con un “Username” y “Nickname” descriptivos.
  - a. En el campo **School** rellenen con “UCSC”
3. Luego entren a la siguiente competencia [Competencia 1](#).
4. **Password:** ucsc

# Reglas Competencia 1 (Evaluada)

1. Con 1 problema resuelto = 30.
2. Con 2 problemas resueltos = 50.
3. Con 3 problemas resueltos = 70.

# Cómo subir una solución

1. Hacer clic en “***Submit***”.
2. En ***Problem***, seleccionar el problema del cual tiene la solución programada.
3. En ***Language***, seleccionar ***Java 1.8.0\_241*** ó ***Java 1.8.0***.
4. En ***Share***, seleccionar ***NO***.
5. En ***Solution***,
  - a. Copiar su código SIN el package.
  - b. Incluir las bibliotecas (import).
  - c. El nombre de la clase principal debe de ser **Main**.

# Ejemplo

Submit

Problem:

D - Problema D

Language:

Java 1.8.0\_241

Share:

Yes

No

Solution:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Collections;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        List<Integer> arr = new ArrayList<>();

        int n = sc.nextInt();

        for (int i = 0; i < n; i++) {
            arr.add(sc.nextInt());
        }

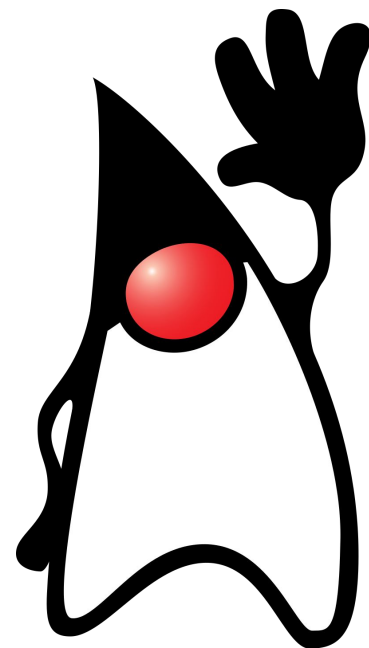
        Collections.sort(arr);

        for (int i = 0; i < arr.size(); i++) {
            System.out.print(arr.get(i) + " ");
        }
    }
}
```

Cancel

Submit





Boris Molina Igor  
Taller de programación II  
[bmolnai@ing.ucsc.cl](mailto:bmolnai@ing.ucsc.cl)