

## UNIDAD TEMÁTICA 1: Introducción a JAVA

### PRACTICOS DOMICILIARIOS INDIVIDUALES #5

#### EJERCICIO #1

Revisar el tutorial de ORACLE, en la parte de tipos enumerados  
(<http://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>)

La declaración “enum” define una clase (llamada tipo enumerado). El cuerpo de esta clase puede incluir métodos y otros campos. En particular, el compilador automáticamente agrega algunos métodos especiales cuando crea un enum. Por ejemplo, tiene un método de valores estáticos que retorna un array que contiene todos los valores del enum en el orden en que fueron declarados.

- 1) Escribe un ejemplo de uso de tal método, y asegúrate de comprender cómo funciona!!
- 2) Teniendo presente el programa que tu Equipo escribió para contar vocales y consonantes en una cierta frase, ¿cómo podrías escribirlo nuevamente utilizando tipos enumerados?

## EJERCICIO #3 Strings.

Revisar el tutorial de ORACLE en <http://docs.oracle.com/javase/tutorial/java/data/strings.html>

Analizar el ejercicio sobre “palíndromo”. Introducir el código fuente en NetBeans y probarlo con diferentes frases. Ver qué sucede al tener letras en mayúsculas y minúsculas, y algunos símbolos de puntuación como “:”, “,” y “;”.

## EJERCICIO #4 Conversión de strings en números

<http://docs.oracle.com/javase/tutorial/java/data/converting.html>

dado el siguiente código fuente:

```
public class ValueOfDemo {
    public static void main(String[] args) {

        // this program requires two
        // arguments on the command line
        if (args.length == 3) {
            // convert strings to numbers
            float a = (Float.value (args[0])).floatValue();
            float b = (Float.valueOf(args[1])).float ();

            // do some arithmetic
            System.out.println("a + b = " +
                               (a + b));
            System.out.println("a - b = " +
                               (a - b));
            System.out.println("a * b = " +
                               (a * b));
            System.out.println("a / b = " +
                               (a / b));
            System.out.println("a % b = " +
                               (a % b));
        } else {
            System.out.println("This program " +
                               "requires two command-line arguments.");
        }
    }
}
```

- 1) Verificar que funciona correctamente e indicar cuál es la salida cuando se invoca con parámetros 13.4 y 66.1
- 2) ¿cómo debería modificarse el código si los parámetros de línea de comando fueran solamente enteros positivos?

## EJERCICIO #5 Conversión de números en strings

<http://docs.oracle.com/javase/tutorial/java/data/converting.html>

Dado el siguiente código:

```
public class ToStringDemo {  
  
    public static void main(String[] args) {  
        double d = 888.51;  
        String s = Double.toString(d);  
  
        int dot = s.indexOf('.');  
  
        System.out.println(dot + " digits " +  
            "before decimal point.");  
        System.out.println( (s.length() - dot - 1) +  
            " digits after decimal point.");  
    }  
}
```

- 1) Indicar cuál es la salida obtenida al ejecutarlo.
- 2) Indicar por qué se imprime cada uno de los datos y la razón de su forma

## EJERCICIO #6 Métodos muy útiles de Strings.

Revisa el tutorial de ORACLE en

<http://docs.oracle.com/javase/tutorial/java/data/manipstrings.html>

Escribe en un papel – archivo txt - qué es lo que hacen los siguientes métodos de String, y, simultáneamente, escribe código bien sencillo que use cada método y muestre en consola el resultado de ejecutarlo (con una string definida internamente para mayor facilidad):

- substring – en todas sus variantes
- split – en todas sus variantes
- subSequence
- trim
- toLowerCase
- toUpperCase
- indexOf – en todas sus variantes, que son muchas y muy poderosas
- lastIndexOf – ídem
- contains
- replace– en todas sus variantes
- replaceAll– en todas sus variantes
- replaceFirst

## EJERCICIO #7 StringBuilder.

Revisa el tutorial de ORACLE en <http://docs.oracle.com/javase/tutorial/java/data/buffers.html>

¿qué es la clase StringBuilder? ¿para qué sirve?

Escribe en un papel – archivo txt - qué es lo que hacen los siguientes métodos de StringBuilder, y, simultáneamente, escribe código bien sencillo que use cada método y muestre en consola el resultado de ejecutarlo:

- Todos los constructores
- setLength
- ensureCapacity
- append
- insert
- delete
- deleteCharAt
- reverse

## EJERCICIO #8.

¿Cuál es la capacidad inicial del siguiente stringBuilder?

```
StringBuilder sb = new StringBuilder("Able was I ere I saw  
Elba.");
```

## EJERCICIO #9.

Considere la siguiente string:

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

- a) ¿qué valor muestra la expresión "hannah.length"?
- b) ¿qué valor es retornado por la invocación del método "hannah.charAt(12)"?
- c) Escribe una expresión que referencie la letra "b" en la string referida por "hannah".

## EJERCICIO #10.

¿Cuán larga es las string devuelta por la siguiente expresión? ¿cuál es la string?

```
"Was it a car or a cat I saw?".substring(9, 12)
```

## EJERCICIO #11.

En el siguiente programa, ¿cuál es el valor de "result" después de la ejecución de cada una de las líneas numeradas?

```
public class ComputeResult {  
    public static void main(String[] args) {  
        String original = "este es el curso de Programación 2";  
        StringBuilder result = new StringBuilder("hola");  
        int index = original.indexOf('a');  
  
        /*1*/ result.setCharAt(0, original.charAt(0));  
        /*2*/ result.setCharAt(1, original.charAt(original.length()-1));  
        /*3*/ result.insert(1, original.charAt(4));  
        /*4*/ result.append(original.substring(1,4));  
        /*5*/ result.insert(3, (original.substring(index, index+2) + " "));  
  
        System.out.println(result);  
    }  
}
```