



UNLP. Facultad de Informática.  
**Algoritmos y Estructuras de Datos**  
**Cursada 2017**

## Práctica Nº 1-B Introducción al lenguaje Java

**Nota: Crea un proyecto llamado "Practica1B" para guardar todas las clases que implementes en esta práctica. Se recomienda crear un paquete por cada ejercicio .**

1. Escriba tres **métodos de clase (static)** que reciban por parámetro dos números enteros (tipo **int**) a y b e impriman todos los números enteros comprendidos entre a; b (inclusive), uno por cada línea en la salida estándar. Para ello, dentro de una nueva clase escriba un método por cada uno de los siguientes incisos:
  - a. Que realice lo pedido con un **for**.
  - b. Que realice lo pedido con un **while**.
  - c. Que realice lo pedido **sin utilizar estructuras de control iterativas** (*for, while, do while*).

Por último, escriba en el método de clase **main** el llamado a cada uno de los métodos creados, con valores de ejemplo.

En su computadora, **ejecute el programa** y verifique que se cumple con lo pedido.

2. Escriba un método de clase que dado un número **n** devuelva un nuevo arreglo de tamaño **n** con los **n** primeros múltiplos enteros de **n** mayores o iguales que 1.  
Ejemplo:  $f(5) = [5; 10; 15; 20; 25]$ ;  $f(k) = \{nk/k : 1..k\}$

Agregue al programa la posibilidad de probar con distintos valores de **n** ingresándolos por teclado, mediante el uso de *System.in*. La clase **Scanner** permite leer de forma sencilla valores de entrada.

**Ayuda:** Como ejemplo de uso, para contar la cantidad de números leídos hasta el primer 42 se puede hacer:

```
import java.util.Scanner;

public class Contador {

    public static int contar42 ( ) {
        Scanner s = new Scanner(System.in) ;
        int cantidad = 1 ;
        while (s.nextInt() != 42) {
            cantidad++;
        }
        return cantidad;
    }
}
```



3. Creación de instancias mediante el uso del operador *new*

- a. Cree una clase llamada **Estudiante** con los atributos especificados abajo y sus correspondientes métodos *getters* y *setters* (*haga uso de las facilidades que brinda eclipse*)
  - nombre
  - apellido
  - comision
  - email
  - direccion
- b. Cree una clase llamada **Profesor** con los atributos especificados abajo y sus correspondientes métodos *getters* y *setters* (*haga uso de las facilidades que brinda eclipse*)
  - nombre
  - apellido
  - email
  - catedra
  - facultad
- c. Agregue un método de instancia llamado **tusDatos()** en la clase **Estudiante** y en la clase **Profesor**, que retorne un **String** con los datos de los atributos de las mismas. Para acceder a los valores de los atributos **utilice los getters previamente definidos**.
- d. Escriba una clase llamada **Test** con el método **main**, el cual cree un arreglo con **2** objetos **Estudiante**, otro arreglo con **3** objetos **Profesor**, y luego recorra ambos arreglos imprimiendo los valores obtenidos mediante el método **tusDatos()**. Recuerde asignar los valores de los atributos de los objetos **Estudiante** y **Profesor** invocando los respectivos métodos *setters*.
- e. Agregue dos breakpoints, uno en la línea donde itera sobre los estudiantes y otro en la línea donde itera sobre los profesores
- f. Ejecute el **Test** en modo debug y avance **paso a paso** visualizando si estudiante o el profesor recuperado es lo esperado.

4. ¿Qué imprime el siguiente programa al ejecutar **main**?

- a. Intente averiguarlo sin ejecutar el programa en su computadora.
- b. Ejecute el ejercicio en su computadora, y compare su resultado con lo esperado en el inciso anterior.

```
public class SwapValores {
    public static void swap1 (int x, int y) {
        if (x < y) {
            int tmp = x ;
            x = y ;
            y = tmp;
        }
    }

    public static void swap2 (Integer x, Integer y) {
        if (x < y) {
            int tmp = x ;
            x = y ;
            y = tmp;
        }
    }
}
```



UNLP. Facultad de Informática.  
**Algoritmos y Estructuras de Datos**  
**Cursada 2017**

```
public static void main(String[] args) {  
    int a = 1, b = 2;  
    Integer c = 3, d = 4;  
    swap1(a,b);  
    swap2(c,d);  
    System.out.println("a=" + a + " b=" + b) ;  
    System.out.println("c=" + c + " d=" + d) ;  
}
```

- c. Inserte un breakpoint en las líneas donde se indica: `y = tmp` y ejecute en modo debug ¿los valores que adoptan las variables `x`, `y` coinciden con los valores impresos por consola?
5. Dado un arreglo de valores tipo **int** se desea calcular el valor máximo, mínimo, y promedio en un único método. Escriba tres métodos de clase, donde respectivamente:
- a. Devuelva lo pedido por el mecanismo de retorno de un método en Java (`"return"`).
  - b. Devuelva lo pedido interactuando con algún parámetro (el parámetro no puede ser de tipo arreglo).
  - c. Devuelva lo pedido sin usar parámetros ni la sentencia `"return"`.