

Práctica 1: Uso de Java y Eclipse

Programación III

Cursada 2013

Todos los ejercicios de la presente práctica deberán hacerse en **Java** dentro de un mismo proyecto llamado “**Prog3_2013**”. Se recomienda realizar cada ejercicio en un paquete diferente, y se recomienda el uso de **Eclipse**.

Además, se recomienda realizar todas las practicas de la presente materia dentro de un mismo proyecto de **Eclipse**, a fin de poder reutilizar código escrito entre diferentes ejercicios y prácticas.

1. **Sin utilizar Eclipse** (por ejemplo en el *block de notas* de Windows), escriba una nueva clase conteniendo el método **main** donde se imprima el mensaje “**Programacion III**”. Desde la terminal/consola/cmd, compile y ejecute el programa mediante el uso de “**javac**” y “**java**”, respectivamente.
2. Escriba tres métodos de clase (función **static**) que reciban por parámetro dos números enteros (**ints**) a, 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, uno por cada una de lo siguiente:
 - 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**).

Tener en cuenta que no necesariamente $a < b$.

Por último, escriba en el método de clase **main** el llamado a cada una de los métodos creados, con valores de ejemplo. En su computadora, **ejecute el programa** y verifique que se cumple con lo pedido.

3. 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: $f(5) = [5, 10, 15, 20, 25]$; $f(k) = \{nk/k : 1..k\}$.
4. Escriba dos métodos de clase donde dado un arreglo de **ints** devuelva la cantidad de elementos diferentes de cero:
 - a) La primera mediante el uso de **for** o **while**.
 - b) La segunda sin utilizar estructuras de control iterativas.

Ejemplo: $f([1, 2, 3, 4, 1, 2, 3, 4, 0]) = 8$.

Invoke los métodos creados desde el método **main**, tomando los valores:

- i. Con valores de ejemplo indicados en el código fuente del programa.
- ii. Con los valores recibidos por el parámetro `args` de `main`.
Investigue el método `Integer.parseInt(x)` para convertir un `String` a un `int`.
- iii. Con valores recibidos por teclado mediante el uso de `System.in`.
La clase `Scanner` permite leer de forma sencilla valores de entrada. Por ejemplo, para contar la cantidad de números leídos hasta el primer 42 se puede hacer:

```
1 public static int contar42() {
2     Scanner s = new Scanner(System.in);
3     int cantidad = 1;
4     while (s.nextInt() != 42) {
5         cantidad++;
6     }
7     return cantidad;
8 }
```

5. ¿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.

```
1 class X {
2     public static void swap1(int x, int y) {
3         if (x < y) {
4             int tmp = x;
5             x = y;
6             y = tmp;
7         }
8     }
9
10    public static void swap2(Integer x, Integer y) {
11        if (x < y) {
12            int tmp = x;
13            x = y;
14            y = tmp;
15        }
16    }
17
18    public static void main(String[] args) {
19        int a = 1, b = 2;
20        Integer c = 3, d = 4;
21
22        swap1(a, b);
23        swap2(c, d);
24
25        System.out.println("a=" + a + " _b=" + b);
26        System.out.println("c=" + c + " _d=" + d);
27    }
28 }
```

6. Dado un arreglo de `ints` 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.
 - c) Devuelva o permita consultar lo pedido sin hacerlo mediante `return` ni por algún parámetro de retorno. No usar variables globales ni `“public static”`.

Por último, escriba en el método `main` el llamado a cada una de los métodos creados, con valores de ejemplo. En su computadora, **ejecute el programa** y verifique que se cumple con lo pedido.

7. Escriba un método que dado un arreglo de `ints` previamente ordenado, devuelva **de forma eficiente** la posición de un elemento concreto indicado por parámetro. En caso de no existir el elemento, el método deberá devolver `«-1»`. El encabezado/*firma* del método es:

```
1 public static int buscar(int [] valores , int elemento) { ... }
```

¿Con qué nombre se le conoce al algoritmo implementado?

8. ⊗ Implemente un método de clase para ordenar un arreglo de `ints` por el método del “Merge-sort”. Investigue el algoritmo en caso de no conocerlo.