# Fundamentos de Programación (ST0242)

## Taller número 4

*La respuesta a este taller se debe enviar, por Eafit InteractivaVirtual, en la fecha y hora que dice la aplicación. El taller se debe desarrollar individualmente. La respuesta a los problemas del punto 1 se agrupan en una clase que se llama* ***Taller4****. Para esta parte se incluye la carpeta con todo el proyecto de BlueJ empaquetada en un archivo zip. Cada problema se soluciona en un método aparte. El primer método se llama* punto1A(), *el segundo se llama* punto1B(), *y así sucesivamente. La respuesta a los problemas del punto 2 se debe entregar en un archivo texto llamado* ***taller4.txt****. Las respuestas al punto 3 se deben entregar en un archivo llamado* ***punto3.pdf****.*

*Para leer un número en Java), puede basarse en el siguiente ejemplo:*

import java.util.Scanner;

public class LeerUnNumero

{

public static void leerNumero() {

System.out.println("Entra un número entero: ");

Scanner in = new Scanner(System.in);

int num = in.nextInt();

System.out.println(num);

}

}

Para recordar el significado del sumatorio () pueden mirar Wikipedia: <https://es.wikipedia.org/wiki/Sumatorio>. Lo mismo para el significado del productorio (): <https://es.wikipedia.org/wiki/Productorio>.

1. Escriba, en Java, la solución para los siguientes retos:
   1. (\*) Un algoritmo que imprima los números divisibles por 5 entre 30 y – 30 (incluidos ambos). Notar que se imprimen de manera descendente.
   2. (\*) Un algoritmo que reciba un entero positivo *n* como parámetro e imprima el resultado de *x*:
   3. (\*) Un algoritmo que reciba dos números positivos *m* y *n* (*n* > *m*) e imprima el resultado de *y*:

(*Hint*: usar ciclos anidados)

* 1. Desarrollar un método para calcular y retornar la **factorial** (*n!*) de un número *n* (mayor o igual que 0) que se recibe como parámetro. Por ejemplo 5! = 120.
  2. Escriba un programa para generar las tablas de multiplicar del 1 al 10. Esto es, la salida del programa debe ser algo así:

1 x 1 = 1

1 x 2 = 2

…

1 x 10 = 10

2 x1 = 2

2 x 2 = 4

…

2 x 10 = 20

…

10 x 10 = 100

* 1. Desarrollar un método **potencia** que reciba una base (tipo double) y un exponente (mayor o igual que 0, tipo entero) y retorne el resultado de la siguiente expresión:
  2. Escriba un programa que acepte un número entero positivo e imprima sus divisores enteros.
  3. Un programa que determine si un número *x* (mayor o igual que 0) que se lee es primo o no (recordar que un número es primo si solamente tiene divisores 1 y el mismo número).
  4. En un programa con menús numéricos, el usuario debe entrar un número entre 1 y 5. Escriba un ciclo que solo deje “pasar” al usuario si entra un número en el rango 1 <= n <= 5. Si el número no es válido, pedirlo de nuevo. Utilice un ciclo MIENTRAS (while () … ), ahora utilice un ciclo REPETIR (do … while()).

*Hint:* Recuerde el problema de la cerca.

* 1. Consultar la expansión de Taylor de la función **coseno**. Haga un programa para calcular el coseno de 60 grados (). Haga un ciclo calculando los términos y acumulándolos. El ciclo termina cuando el valor absoluto del término que se acaba de calcular sea menor que 0.0001 (ver: <https://es.wikipedia.org/wiki/Serie_de_Taylor>). Utilice el método para calcular el **factorial()** y el método para la potenciación **potencia()** que escribió anteriormente. Para el valor absoluto, puede utilizar la función Math.abs().
  2. Consulte el algoritmo de **Euclides**: <https://en.wikipedia.org/wiki/Euclidean_algorithm>. Impleméntelo en java y explique lo que el programa hace brevemente en un comentario al comienzo del método.
  3. Consulte la secuencia de **Fibonacci** (https://en.wikipedia.org/wiki/Fibonacci\_number). Implemente un método en Java que reciba un número positivo *n* (mayor o igual que 3) e imprima los primeros *n* números de la secuencia de Fibonacci. Se asume que el primer número es 0 y el segundo número es 1.
  4. Escriba un método que use valor centinela. El método le debe pedir al usuario que entre números *negativos*. Una vez que el usuario entre un número *positivo*, el programa debe mostrar el valor máximo y mínimo entre los números negativos introducidos.
  5. Escriba un método llamado **sumaDados**() que acepte un Scanner como parámetro para leer desde la consola. El método debe pedirle al usuario una suma deseada (un número entre 2 y 12) y luego simular que se lanzan 2 dados, usando la función Math.random() de Java, hasta que se obtenga la suma deseada. El siguiente es un ejemplo del diálogo con el usuario:

Suma deseada: 9

4 y 3 = 7

3 y 5 = 8

5 y 6 = 11

6 y 3 = 9

* 1. Escriba un método que reciba como parámetro un número n (mayor que 0) y escriba un patrón como se describe a continuación.

Para n = 4:

...4

..3\*

.2\*\*

1\*\*\*

Para n = 5:

....5

...4\*

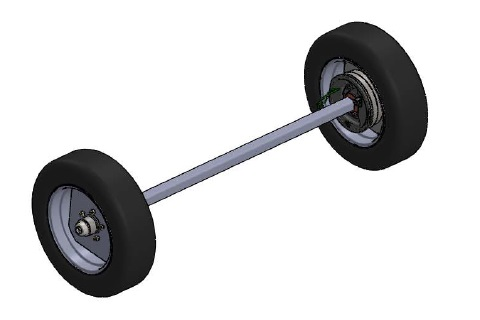
..3\*\*

.2\*\*\*

1\*\*\*\*

(*Hint*: use ciclos anidados)

* 1. ¿Se imagina que el tiempo retrocediera? Podemos hacer un simulador de reloj que muestre las horas como si el tiempo retrocediera, es decir, 12:00 11:59 11:58 11:57 … 0:03 0:02 0:01 0:00. Escriba un programa que imprima el tiempo retrocediendo, comenzando desde las 12:00.
  2. Escriba un programa que calcule la distancia recorrida por un vehículo. Se conoce cuántas veces giró el eje y también se conoce el radio de las llantas. El programa debe determinar la distancia recorrida.



* 1. Una empresa que se dedica a la venta de combustible necesita un programa para gestionar las facturas. En cada factura aparece: el código del artículo, la cantidad vendida en litros y el precio por litro. Se pide, para 5 facturas introducidas: Facturación total, cantidad en litros vendidos del artículo 1 y cuantas facturas se emitieron de más de $500000.
  2. Necesitamos mostrar un contador con 5 dígitos, que muestre los números del 0-0-0-0-0 al 9-9-9-9-9, con la particularidad que cada vez que aparezca un 1 lo reemplace por una I y que cada vez que aparezca un 3 lo sustituya por una E.

1. Incluya las respuestas a los siguientes puntos en un archivo texto llamado taller4.txt.
   1. Convierta el siguiente programa que usa ciclo for en un programa que solo usa ciclos while:

for(int i = 1; i <= 5; i++) {

for(int j = 1; j <= 10; j++) {

for(int k = 1; k <= 15; k++) {

System.out.println("\*");

}

System.out.println("<>");

}

System.out.println();

}

* 1. Considere el siguiente método:

public static void mystery(int x) {

int y = 1;

int z = 0;

while(2 \* y <= x) {

y = y \* 2;

z++;

}

System.out.println(y + " " + z);

}

Para cada llamado, indique la salida del método anterior

mystery(1)

mystery(6);

mystery(19);

mystery(39);

mystery(74);

1. Consultas: Realice las siguientes consultas en medios físicos o electrónicos en la biblioteca de la Universidad o por Internet. Preferiblemente el texto de la asignatura. MUY IMPORTANTE: No olvide citar las fuentes.

* En qué consiste la Programación Orienta a Objetos
* Qué es una clase y qué es un objeto
* Qué son atributos y qué son métodos
* Construya una clase con 3 atributos y 4 métodos