# Fundamentos de Programación (ST0242)

## Taller número 6

*La respuesta a este taller se debe enviar, por Eafit InteractivaVirtual, a más tardar en la fecha indicada en la aplicación. El taller se debe desarrollar individualmente. Cada problema se soluciona en un método aparte. Usted va a construir varias clases. Las clases se deben crear al interior de un mismo proyecto llamado Talle6. Debe subir a Eafit el directorio que contiene el proyecto de BlueJ empaquetado en un archivo zip.*

*Nota: En los puntos 1 al 6, no es necesario usar recursión. En los puntos 7 en adelante SI se debe usar recursión.*

*La respuesta a los puntos 1 al 5 se debe incluir en una clase llamada* Metodos*.*

1. Escriba un método llamado printPowersOf3(int n) que acepte un número máximo como argumento e imprima cada potencia de desde 30 (1) hasta 3n, inclusive. A manera de ejemplo:

* printPowersOf3(3) imprime 1 3 9 27
* printPowersOf3(10) imprime 1 3 9 27 81 243 729 2187 6561 19683 59049

1. Escriba un método llamado printGrid(int numFilas, int numCols) que acepte dos enteros que representen el número de filas y columnas e imprima una matriz desde 1 hasta (filas \* columnas) de manera descendente. Naturalmente, no se pueden utilizar matrices de Java, ya que no se ha visto esta estructura. A manera de ejemplo:

* printGrid(4, 6) imprime:

1 5 9 13 17 21

2 6 10 14 18 22

3 7 11 15 19 23

4 8 12 16 20 24

1. Escribe un método llamado invertirNombre(String s) que le pregunte al usuario su nombre y luego imprima el nombre al revés. A manera de ejemplo:

invertirNombre (“juanita perez”) retorna

zerep atinauj

1. Escriba un programa para dibujar árboles de navidad. El método recibe dos parámetros: el número de segmentos del árbol y la altura de cada segmento. Por ejemplo, el siguiente árbol tiene 3 segmentos, cada uno de altura 4:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*

\*

\*\*\*\*\*\*\*

Y el siguiente tiene 2 segmentos, cada uno de altura 5:

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*

\*

\*\*\*\*\*\*\*

1. Los siguientes métodos se deben hacer en una clase llamada PorValorYPorReferencia:
   1. Para demostrar el paso de parámetros por valor, escriba un método llamado modificarEntero(int n) que reciba un parámetro formal n y modifique el valor. El método se llama desde un método llamado testModificarEntero(), en el cual se invoca el método anterior. En éste método se imprime un entero antes y luego de invocar modificarEntero(), para mostrar que el valor del parámetro real ***no*** se modifica.
   2. Escriba dos métodos similares al punto anterior, pero para demostrar que un valor tipo double también se pasa por valor.
   3. Ahora cree una clase llamada Wrapper con un solo atributo entero con visibilidad public. Le puede dar el nombre que quiera. Escriba un método llamado modificarObjeto(…) que reciba una referencia a un objeto del tipo Wrapper. Dentro de este método se debe modificar directamente el atributo del objeto (lo cual **si** se puede hacer porque la visibilidad es public). Ahora escriba un método llamado testModificarObjeto(), en el cual se invoca el método anterior. Imprima el valor del atributo de una instancia del objeto antes y después de invocar al método modificarObjeto(), para mostrar que si los parámetros son referencias a objetos, el paso de parámetros tiene una semántica diferente (paso de parámetros por referencia).
2. Escriba un método que imprima la(s) solución(es) reales de una ecuación de segundo grado (ver la definición y la forma de solucionarla en el siguiente link: <https://es.wikipedia.org/wiki/Ecuaci%C3%B3n_de_segundo_grado>). Si no existen soluciones reales, el método debe informarle al usuario.
3. Escriba un método recursivo llamado mcd(int m, int n) que calcule el máximo común divisor (MCD) de dos enteros positivos que se reciben como parámetros. Si m >= n una función recursiva para MCD es la siguiente:

* MCD = m (si n es igual a 0)
* MCD = MCD (n, m % n) (si n no es igual a 0)

Notar que la definición anterior usa recursión en la función MCD. Si el usuario ingresa un valor para m que es menor que n el método debe intercambiar los valores.

1. Escriba un método recursivo que permita invertir un número entero que se recibe como parámetro. Esto es: si se recibe el número 123, el resultado debe ser el número 321. El encabezado del método debe ser como el siguiente: public static int invertir(int n). En la implementación del método solo se deben usar operaciones aritméticas, no se debe utilizar un String.
2. Resuelva los puntos de **Recursion-1** de Codingbat citados a continuación. Para que se puedan calificar, usted debe crear una cuenta así: <codigoDeEstudiante>st0242s20181g<grupo>t6@hotmail.com (no es necesario que la cuenta exista). Y debe agregar al profesor: st0242s20181g<grupo>t6@hotmail.com. Si usted no realiza este proceso, esta parte del taller no podrá ser calificada. Para agregar al profesor, debe entrar por la opción “prefs” y luego “share to”.
   1. Fibonacci
   2. triangle
   3. sumDigits
   4. count7
   5. count8
   6. powerN
   7. countX